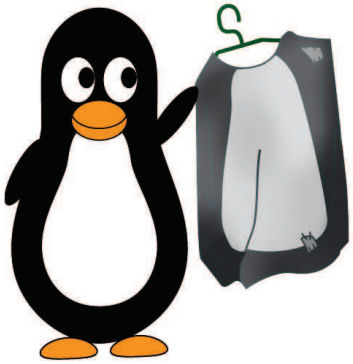
Beköszöntő

Linuxvilág



Szy György
főszerkesztő

A Pingvin új ruhája

Már-már a hazai hivatalokat is megszégyenítő sebességgel, hosszadalmas háttérmunka után, végre elérkezett az idő, hogy a világ elé tárjuk a megújult Linuxvilágot! A változtatások valószínűleg mindenkinek szembeötlőek lesznek, bár bevallom, azon véleményemet, miszerint is jelenjen meg a lapban minél több szép lány, csak részben tudtam érvényesíteni. Fontosabb cél volt ugyanis, hogy olvasóink igénye szerint elsősorban a tartalomra helyezzük a hangsúlyt: kevesebb erősen szakmai cikk, több érdekesség, több hazai anyag.

Természetesen az új laptartalom mellett szerettük volna a linuxos közösség által tiszteletnek örvendő nemes tityitotyi állatkák előtt is tisztelegni, amihez a legjobb megoldást a Budapesti Állatkert szolgáltatja. Itt lakik ugyanis tizenegy darab pingvinke. Szerkesztőségünk ünnepélyesen örökbefogad egy pingvint, csak még azt nem tudjuk, hogy gyermekünk Alex legyen-e, vagy Tóbiás (akiről nemrég kiderült, hogy lehet, hogy lány). Részletesebben a 17. oldalon!

Az új grafikai témájú sorozatok mellett (Blender, POVRay, SVG) a mostani számban közöljük le a Linux Journal csapata által épített néma erőműről szóló cikket is, a 24. oldalon. (Halló, munkáltatók! Ma már nem luxus a csendes iroda!) És ha van egy sznobok számára is megfelelő csúcsgépünk, hát futtassunk rajta húsz-harminc másikat! Ennek megoldásához három módszert hasonlítunk össze, a XEN, a vmWare és a Qemu programok személyében (az összeállítás a 34. oldalon kezdődik).

A temérdek változtatásról természetesen szeretnénk minél több véleményt kapni, úgyhogy usgyi, a lap átolvasása után mindenki küldjön egy Nekemnagyon-tetszik-levelet címünkre: szerkesztoseg@linuxvilag.hu

Kellemes olvasást kívánunk!

Hírek

A kontinensre lép a LinuxIT

A ***LinuxIT*** bejelentette, hogy többnyelvű angliai szolgáltató- és támogatási központjával belép a kontinentális európai piacra is. Anglia vezető független linuxos rendszerintegrátora és szolgáltató cége az ***LinuxIT Europe*** révén elsősorban a heterogén rendszereket üzemeltető, illetve a meglévő rendszerüket linuxos elemekkel bővíteni kívánó vállalatoknak kínálja szaktudását. A ***LinuxIT*** a tanácsadás mellett képzéseket is biztosít.
➲ www.linuxit.com

Origami holnapra

A ***Graphtec CraftROBO*** névvel kisméretű, főként otthoni jellegű használatra alkalmas vágógépet mutatott be. A kinézetét tekintve az egyszerűbb tintasugaras nyomtatókra emlékeztető gép képes a betöltött A4-es méretű papírlapból a kívánt alakzatot kivágni, illetve a megadott sablon alapján a későbbi hajtogatást segítő bemetszések készítésére is alkalmas. A gép legfeljebb 0,1 mm vastagságú anyagok kezelésére képes, ezen a határon belül azonban többféle papírtípust vagy éppen fóliát is használhatunk. A modelleket a mellékelt ***ROBO Master*** alkalmazással készíthetjük el, de ha van egy kis szerencsénk, akkor más rajzolóprogram alá is találunk megfelelő beépülő modult; a gyengébb fantáziájúak pedig a gyári sablonok közül válogathatnak. A gép ára körülbelül 400 euró.
➲ http://www.graphteccorp.com/craftrobo/

Igen is meg nem is

Az ***Apple*** súlyos szentségtörést követett el, amikor szakított az egygombos egerek évtizedes hagyományával. Pontosabban nehéz volna definiálni, hány gombja van a ***Mighty Mouse***-nak, hiszen – ha már változtatott – az ***Apple*** most sem ragadt le a hagyományos megoldásoknál: az új rágcsáló rendelkezik egy tetszőleges irányú gördítésre, léptetésre és kattintásra egyaránt alkalmas gombbal, a hagyományos egerek gombjainak helyén egy érintésérzékeny területtel, továbbá oldalsó, nyomásra érzékeny részekkel. Egyes elemeinek működését a tulajdonos kívánsága szerint programozhatja, így gombtalan egere akár négygombosként is működhet, illetve balkezes használatra is könnyedén átállítható. A mozgásokat optikai módon észleli, a géphez pedig ***USB*** kábellel csatlakozik, így nemcsak ***Mac OS X***, de más operációs rendszerek alatt is használható. A ***Mighty Mouse*** ára 49 dollár, vagyis közel tízezer forint – azért valljuk be, ennyi pénzért kevesen vásárolnak egeret.
➲ www.apple.com

Kivetítő hat alapszínnel

A japán ***Texas Instruments*** bemutatta legújabb, ***BrilliantColor*** technológiát alkalmazó, digitális fényfeldolgozó (***DLP***) kivetítőjének prototípusát. Az új megoldás legfontosabb jellemzője, hogy a magas fényerő megőrzése mellett immár hat alapszín – vörös, zöld, kék, sárga, magenta és cián – megjelenítésére képes, így minden eddiginél élethűbb, szebb színeket képes a vászonra varázsolni. Az újfajta kivetítők gyártását a jövő év elején kezdik meg.

Motorola-Oakley Razrwire

A ***Motorola*** és a felső kategóriás napszemüvegek gyártó ***Oakley*** bemutatták az első napszemüvegre szerelhető ***Bluetooth*** modult. A nehezen kimondható nevű ***Razrwire*** a napszemüveg szárára – kívánság szerint bal vagy jobb oldalra – illesztve tulajdonképpen hagyományos ***Bluetooth*** headsetként működik. A ***Razrwire*** a ***Bluetooth 1.1***-es és ***1.2***-es változatát támogatja, két hangerő-szabályozó és egy darab multifunkciós gombjával minden fontosabb hívásfunkció elérését biztosítja. A modul beépített akkumulátorral rendelkezik, készenléti ideje 100 óra, beszélgetési ideje pedig 5 óra. A ***Razrwire*** a ***Cingular*** szolgáltatótól (➲ ***www.cingular.com***), a ***Motorolától*** (➲ ***www.hellomoto.com***) és az ***Oakley***-től (***www.oakley.com***) rendelhető meg, igazi divatcikkhez méltó, 300 dolláros áron.
➲ http://oakley.com/about/razrwire/

Nanoszalag

FUJIFILM

A jövő szelét hozza magával a ***Fuji*** legújabb, ***Fujifilm 3592 WormTape*** szalagja, mely már nanotechnológiai megoldásokkal készül. A ***Fuji*** rendkívül vékony ***Nanocubic*** bevonata különösen kisméretű részecskékből áll, ezek alkalmazásával minden korábbinál nagyobb adatsűrűséget sikerült elérni, miközben a zajt és a jel-zaj arányt hatékonyan tudták csökkenteni. Az új szalagok natív kapacitása 300 GB, tömörítéssel akár 900 GB rögzítésére is alkalmasak lehetnek, 110 MBps átviteli sebesség mellett. A szalagos adattárolás bármennyire is egy letűnt kor eszközének tűnik, várha-

tóan még jó ideig nem tűnik el a számítógépközpontok hétköznapjaiból, ugyanis a világ különböző tájain egyre komolyabb a törekvés arra, hogy a hatóságok a távközlési, banki, internetes és egyéb adatok megőrzésére a lehető legszélesebb körűen és a lehető leghosszabb időre kötelezzék a szolgáltató vállalatokat.
➲ www.fujifilm.com

Csak az ujjad ne veszítsd el

A biometriai készülékeket gyártó *Bionopoly* újdonsága a *Computer-on-a-Stick (COS)* – kb. számítógép *USB* kulcson – elgondolás alapján készülő, 256 MB kapacitású, *LCD* kijelzővel és ujjlenyomat-olvasóval is ellátott, előtelepített *Debian Linuxot* tartalmazó *USB* kulcs. A *COS* a gyártó szerint bármilyen *x86* alapú számítógépen alkalmas a rendszerindításra, és mivel számos felhasználói programot is tartalmaz, kiválóan megfelel egyszerű, hordozható munkakörnyezet létrehozására. A *COS* természetesen a felhasználó dokumentumainak tárolására is alkalmas, ezt titkosított vagy szabad hozzáférést biztosító módon képes megoldani, vagyis a hagyományos *USB* kulcsokhoz hasonló módon is használható. A *COS* ára 199 dollár, de aki nélkülözni tudja az ujjlenyomat-olvasót, az 149 dollárért is megszerezheti.
➲ www.bionopoly.com

Memóriából gyorsabb

Talán nincs már messze az az idő, amikor az átlagos felhasználók is megszabadulhatnak a sérülékeny, magas fordulattal pörgő korongokat, ide-oda csapkodó fejeket tartalmazó merevlemezes meghajtóktól. A *Samsung* már dolgozik az adattároló eszközök következő generációján, a szilárdtest-memóriás meghajtókon, amelyek *Flash* memóriát fognak alkalmazni az adatok tárolására. A mozgó alkatrészek eltűnésének előnyei egyértelműek: megszűnik a fizikai hatásokra való érzékenység, csökken a főleg a hordozható gépeknél fontos fogyasztás, és várhatóan a számítógépek rendszerindítási ideje is jelentősen lerövidül. Természetesen a *Flash* memória egyelőre – és várhatóan még jó ideig – drágább, mint a hagyományos merevlemez, ezért elsőként a két megoldást egyesítő, alig vagy egyáltalán nem drágább, 64-128 MB *Flash* memóriát tartalmazó hibrid meghajtók elterjedésére lehet számítani. Ezek fejlesztésében a *Microsoft* és a *Samsung* szorosan együttműködnek, az elképzelések szerint az operációs rendszer a Flash memóriát egyfajta szuper-gyorsítótárként fogja használni. Földhöz ragadtabb, de megfizethető és rendkívül praktikus kiegészítő a *Gigabyte PCI* foglalatba illeszkedő *iRAM* memóriakártyája. Hátránya, hogy folyamatos áramellátást igényel, így adatok hosszú távú tárolására kevésbé alkalmas, ám például munkafájlok ideiglenes elmentésére és gyors visszaolvasására, illetve a merevlemezek terhelésének csökkentésére kiválóan alkalmazható.

Zen Vision

A *Creative* bemutatta legújabb *MP3* lejátszóját, a *Zen Visiont*. Az *MP3* lejátszó megnevezés már korántsem fedi le a legújabb készülék képességeit, hiszen 3,7″-os, színes, *640x480* képpontos felbontású kijelzőjével a Zen Vision fényképek és mozgóképek megjelenítésére is képes. A *Zen Vision* 30 GB-os merevlemezzel rendelkezik, amely akár tizenötezer dal, több tízezer fénykép és legfeljebb 120 órányi videóanyag tárolására alkalmas. Lejátszóként jó néhány formátumot támogat – köztük található a *DivX* és az *Xvid* is –, segítségével különféle internetes zeneáruházakból is vásárolhatunk, illetve beépített *FM* vevője révén rádiófelvételek készítésére is használható. A fájlokat *CompatFlash* kártyáról is képes átmásolni, külön megvásárolható kiegészítő révén pedig további kártyatípusokat is tud olvasni, így a viszonylag szűkös kapacitással ellátott digitális fényképezőgépek mellé ideális társ. A készülék kétféle, fehér és fekete színben kapható, belső akkumulátorával 4,5 órás üzemidőre képes, súlya 240 gramm, ára pedig 400 dollár.
➲ www.creative.com

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

Mi újság a rendszermag fejlesztése körül?

A *Linux* rendszermag fejlesztése körül nemrég káosz alakult ki, amikor *Larry McVoy* véglegesen úgy döntött, hogy visszavonja az ingyenes *BitKeeper* licencet – ezzel a lépéssel már többször is fenyegetőzött. Alig néhány nap kellett csak, és *Linus Torvalds* segítők egész csapatától körülvéve, a nulláról indulva elfogadható alternatívát készített. A *git* fájlrendszer *Linus* agyszüleménye, egy alacsony szintű, rendkívül gyors tartalomkövető, ami a meglévő változatkezelő megoldásokhoz viszonyítva teljesen idegen elgondolásnak látszik. Gyakorlatilag átlátszó, célja az, hogy egy különféle parancsfájlokból álló, a szolgáltatásait kiaknázó réteg alatt fusson. Az alaprendszer fölé parancsfájlok írásával bárki készíthet új *gites* felhasználói réteget. *Petr Baudis* és sokan mások komoly munkát fektettek a *Cogitoba*, egy a *githez* készült felhasználói felületbe, a jelek szerint *Linus* is ezt választja a rendszermag folyó fejlesztéseihez. A különféle webes felületek és külső eszközök szinte naponta jelennek meg hozzá. *H. Peter Anvin* példásan gondozta a *kernel.org*-ot, naprakészen tartva a *git* gyűjteményeket, miközben tárhelyszolgáltatásról és úgy általában, mindenről gondoskodott. Ennek ellenére a *kernel.org* az utóbbi időben eléggé leterheltté vált, a világ minden részéről érkező kérések kiszolgálásához hatalmas sávszélességre lett volna szükség. Ez alkalommal a *Hewlett-Packard* dobott mentőövet, két nagyteljesítményű gép formájában. Mindkét gépről a *kernel.org* fog futni, a kérések megosztása *DNS* alapon történik. A bővítésnek köszönhetően a webhely válaszidői sokat javultak, és a hozzájárulások feltöltésének sebessége is nagyságrenddel nőtt. Az egyetlen gond már csak az, hogy a *DNS* alapú elosztás miatt nehéz összeállítani a hálózati forgalmi statisztikákat. *Joel Becker* elkészítette a *ConfigFS*-t, egy újabb felületet a rendszermag belső világához. A cél ezúttal valami parancsfájlokból is kezelhető, teljes mértékben olvasható dolog létrehozása volt. A *SysFS* azonban már létezik, és nagyon hasonló feladatot lát el, vagyis nem tiszta, hogy a *ConfigFS* valóban fog előnyöket kínálni, vagy csak többen leszünk vele. Mindezek a fájlrendszer alapú felületek azon elvárás nyomán fejlődtek ki, hogy a Linux végre megszabaduljon az egyébként kiváló elődöktől fennmaradt *ProcFS*, */dev* és *ioctl* terhes örökségétől. Ám ha az új alternatívák nem képesek megfelelni az elvárásoknak, akkor a *SysFS*, az *udev* és most a *ConfigFS* is csak régről örökölt vackok maradnak, melyeket a rendszermag fejlesztői a következő néhány év során töretlenül utálni fognak.

A *FUSE (Filesystem in USErspace, fájlrendszer felhasználói térben)* fejlesztői most vagy rendet tesznek, vagy fokozzák a fejetlenséget. Szeredi Miklós készített néhány foltot a felhasználói felülethez, így az azokon a gépeken, amelyek mindkét módot ismerik, a 32 és a 64 bites működést egyaránt támogatja. Számos előnye mellett a foltozás megtöri a felhasználói felület visszirányú kompatibilitását. A *FUSE* már bekerült *Andrew Morton -mm* fájába, így a folt szenvedéssel teli, mégis szükséges lépés a hivatalos rendszermag felé vezető úton – vagy éppen a végső bukás és az *Andrew* ágából való kikerülés felé. Majd meglátjuk. A dolgok legutóbbi állása szerint a *FUSE* fejlesztői jól haladnak afelé, hogy elhárítsák *Linus* legfontosabb ellenvetéseit, akinek a felhasználói térben futó fájlrendszerekkel kapcsolatos aggályait egyre inkább sikerül eloszlatni.

Az *open-iscsi* és a *linux-iscsi* tervezetek fejlesztői nyilvános vitájukat követően a két tervezet egyesítése mellett döntöttek. Műszaki okokból a két csoport úgy határozott, hogy az open-iscsi kódbázisára kezdenek építkezni, ez a tervezet ugyanis optimalizált be- és kiviteli útvonalakkal és alaposan letesztelt *iscsi-sfnet* összetevőkkel rendelkezik a vezérlőfelülethez és a felhasználói térben futó összetevőkhöz. Az *open-iscsi* alváltozat gyűjteménye továbbra is használatban marad, legalábbis egyelőre. A két tervezet a közös cél érdekében való egyesítése rendkívül örvendetes fejlemény. Remélhetőleg a *linux-iscsi* csoport résztvevőinek nagy része tovább folytatja a munkát, korábbi eredményeik pedig a másik kódbázisra való áttérés ellenére nem vesznek feledésbe.

Randy Dunlap vállalata fel a rendszermag hálózati beállításainak megváltoztatására irányuló erőfeszítések vezetését. Ronda egy feladat, sok esetben ugyanis nem könnyű eldönteni, hogy a hierarchia szervezése hogyan volna a legcélszerűbb. Adott összetevő vajon illesztőprogramnak vagy protokollnak számít? Minden illesztőprogramot egy csoportba kellene sorolni, vagy a csoportosítás során figyelembe kellene venni, hogy az egyes illesztőprogramok milyen alrendszerhez kötődnek? *Randy* vett egy nagy lélegzetet, majd megtette az első lépést, és megadta a választ néhány kényes kérdésre. Rövid idő alatt jó néhány támogatója akadt. Jó adag civakodás és önvizsgálat, valamint némi találgatás által úgy tűnik, fokozatosan ki fog alakulni a hálózati beállítások új arculata. Az új arculat részeit – kisebb-nagyobb megrázkódtatások árán – a 2.6-os sorozat újabb tagjaiban fogjuk megtalálni.

Zach Brown

Linux Journal 2005., 136. szám

Szabad Szoftverek Világnapja 2005

A tavalyi évhez hasonlóan idén is *Szegeden* kerül megrendezésre a *Szabad Szoftverek Világnapja*, amelynek elsődleges célja a szabad szoftverek népszerűsítése. A találkozó programja még kialakulóban van, de alapvetően olyan előadásokat tervezünk, amelyek mind a „nagy *Linux* guruknak", mind az érdeklődő kezdőknek érdekes lehet. A belépés természetesen ingyenes, a programot az előadásokon túl további érdekességek fogják színesíteni, amiről bővebb információ majd a honlapunkon lesz olvasható. A rendezvény szervezése most folyik, akinek van kedve hozzá és csatlakozna akár szervezőként, akár előadóként szívesen vesszük, továbbá minden ötletet is örömmel látunk a levelezőlistánkon. A megvalósítás színvonalának növeléséhez szponzorok jelentkezését is várjuk.

Tervezett időpont:
2005. szeptember 25. vasárnap

A tavalyi megnyitó beszédet dr. Dévényi Károly tanár úr tartotta, az otthont adó intézményt képviselve

Honlapunk: http://www.inf.u-szeged.hu/opensource/sfd/
Tavalyi honlap: http://www.inf.u-szeged.hu/opensource/events.php
Tavalyi képek: http://www.inf.u-szeged.hu/opensource/sfd2004-pics/
Feliratkozás a levelezőlistánkra: http://www.sed.hu/cgi-bin/mailman/listinfo/sfd

A rendezvényre készített GNU/Linux-os pólók is népszerűek voltak

A közönség

Mindenki „betölthetett" egy nyílt forráskódú süteményt (forrás és bináris)

Beszámoló a belgrádi szoftverhonosítási konferenciáról

Május végén az alábbi levél érkezett az FSF.hu Alapítvány e-mail címére:

Kedves kollégák!
Vedran Vučić vagyok, a szabad szoftverek népszerűsítésével foglalkozó Linux Központ nevű civil szervezet elnöke. Konferenciát szervezünk a szabad szoftverek és dokumentációk fordítása, illetve honosítása témakörében. A konferenciát 2005. július 1. és 3. között, Belgrádban rendezzük meg. A meghívott résztvevők Szerbiából, Macedóniából, Boszniából, Horvátországból, Romániából és Bulgáriából érkeznek. Lehetőség van két fő fogadására Magyarországról is. Az utazás, szállás és étkezés költségeit mi álljuk. Kérjük, jelelezzétek, ha érdekel a konferencia. Reméljük, hogy a leendő együttműködésünknek köszönhetően a szerbiai magyarok kényelmesebben tudják majd használni a szabad szoftvereket.

Sokkal több konkrétum több levélváltás után sem derült ki a rendezvényről, de úgy gondoltuk, ha minden költséget állnak, nyugodtan elmehetünk. A két magyar résztvevő ***Daczi László*** a ***Magyar Linux Dokumentációs Projekt*** (➲ ***tldp.fsf.hu***) vezetője és a ***TheOpenCD*** (➲ ***theopencd.fsf.hu***) honosítója, illetve ***Tímár András*** volt, aki a ***Mozilla*** (➲ ***mozilla.fsf.hu***) és az ***OpenOffice.org*** (➲ ***hu.openoffice.org***) honosításával foglalkozik. A helyszínre érkezve sikerült többet is kiderítenünk. A konferenciát az ***Open Society Institute (Soros Alapítvány)*** támogatta. A konferencia célja a térség országaiban élő, szabad szoftverek, illetve dokumentációk fordításával foglalkozók tapasztalatcseréjének megszervezése volt.
Minden ország küldöttsége beszámolt a honosítások terén elért eredményeiről, illetve nehézségeikről. Mi két előadást is tartottunk, az egyiket az általános magyarországi helyzetről (mit értünk el, mik a terveink), a másikat ***Daczi László*** készülő ***s3po*** nevű fordítássegítő programjáról. Erről a programról annyit érdemes tudni, hogy ***Windows*** és ***Linux*** alatt egyaránt fut (***Delphi/Kylix*** feljesztőkörnyezettel készül), és elsősorban a ***Linux HOGYAN*** dokumentumok fordítására ad a jelenleginél sokkal jobb megoldást. Megvalósítja a számítógéppel segített fordítás világában egyre inkább terjedő nyílt szabványokat (például ***XLIFF***), így az eszköz együttműködése más, akár nyílt akár zárt kódú programokkal zökkenőmentes lehet. Az ***s3po*** programról további információ a ➲ ***http://tldp.fsf.hu/s3po/*** címen olvasható.
Anélkül, hogy részletesen ismertetném az egyes előadásokon elhangzottakat, összefoglalom az általános tapasztalatokat, és kitérek a legérdekesebb dolgokra.
Minden országban célul tűzték ki a teljesen honosított, szabad munkaasztali környezet előállítását. A legtöbben rendelkeznek honosított ***GNOME***-mal és ***KDE***-vel, sokan végeztek a ***Mozilla***-család és az ***OpenOffice.org*** honosításával is. Honosított súgók és dokumentációk tekintetében szerényebb volt a felhozatal. A magyar eredményeket (például ***Mozilla*** súgó, ***Firefox*** súgó, készülő ***OpenOffice.org*** súgó, több tucat lefordított ***Linux HOWTO***) a legtöbben irigykedve és csodálattal hallgatták. Igaz, véleményem szerint ***Magyarországon*** jóval előbb elkezdődtek a honosítási munkák, és tekintve a többi országban tapasztalható lelkesedést, egy-két év alatt ők is elérik mindazt, amit mi.
A legtöbb országban szerveztek már „fordítómaratont". Az ***FSF.hu*** 2002 februárjában szervezett ilyent az ***OpenOffice.org*** felhasználói felületének, majd másfél évvel később a részletes tippjeinek lefordítására. Ilyenkor egyszerre akár 100 ember is fordít egy hétvégén át. ***Horvátországban*** nagyon bevált ez a módszer, már 10 ilyen rendezvény volt, és a munkát egy nagyon profi webes fordítássegítő rendszer támogatja (➲ ***lokalizacija.linux.hr***).
Ez a rendszer hasonló azokhoz, amelyeket itthon használtunk, de tudása nagyobb, kiforrottabb és használata kényelmesebb. Magyarországi bevezetésének fő akadálya, hogy az egész rendszert horvátul írták. Inkább szerb, bosnyák és macedón kollégáink érdeklődésére tarthat számot, bár elhangzott egy halvány ígéret az angol változat elkészítésére. A fordítómaratonok tapasztalatai más országokban is hasonlók, mint nálunk. A résztvevők egyetértettek abban, hogy nagy mennyiségű anyag nyersfordítása állítható elő rövid idő alatt. Sok esetben más lehetőség egyáltalán nem adódik nagyobb lélegzetű munkák elvégzésére. A nyersfordítás kijavítása mindenhol szűk keresztmetszetet jelent, és megfogalmazódott, hogy nem kell idegenkedni attól sem, hogy a javítási munkálatokat pénzért végezzék el erre felkészült cégek vagy személyek. Szintén megfogalmazódott, hogy a szabad szoftverek, illetve nyílt forrású szoftverek terjedésében érdekelt nemzetközi cégeket (***Sun***, ***Novell***) minden országban meg kell keresni, és segítséget kell

A konferencia résztvevői
Álló sor balról jobbra: Filip Stojanovski (mk), Hrvoje Spoljar (hr), Tímár András (hu), Santja Kemal (ba), Robert Sedak (hr), Daczi László (hu), Jordanka Pertusevska (mk), Vladimir Petkov (bg), Daniel Secăreanu (ro), Marko Milenović (sr), Ivan Jelić (sr), Novica Nakov (mk)
Ülő sor balról jobbra: Toplica Tanasović (sr), Eddy Petrisor (ro), Nikola Kotur (sr), Ahmet Mulalić (ba), Misu Moldovan (ro), Vedran Vučić (sr)

tőlük kérni a honosítási projektekhez. Ilyen együttműködés ***Magyarországon*** már létezik, de a térség más országaiban még nem jellemző.

A szabad szoftveres élet ***Szerbiában*** nagyon pezsgő. A szokásos dolgokon (***KDE***, ***GNOME*** stb. lefordítása) felül kiemelném az ***FSN Serbia*** tevékenységét. Nevük (***Free Sotware Network***) nem véletlenül hasonlít az ***Összefogás a szabad szoftverek elterjesztésére Alapítvány (➲ www.fsn.hu)*** nevére. Szoros a kapcsolat a két szervezet között, többek közt az ***FSN Serbia*** kiszolgálója (➲ ***www.fsn.org.yu***) is ***Szegeden*** üzemel a magyar ***FSN*** jóvoltából. Az ***FSN Serbia*** leglátványosabb terméke talán a ***GNUzilla*** című ingyenes, ***PDF*** formátumban kiadott, szerb nyelvű, szabad szoftveres magazin. Eddig 7 szám jelent meg, tehát nem egyszeri fellángolásról van szó.

Az előadások közti szünetekben, illetve az étkezések alkalmával kötetlen beszélgetések során megtárgyaltuk, hogy milyen technikai nehézségekkel kell szembenéznie a különféle nyelvek fordítóinak. Kiderül, hogy mi, magyarok igen szerencsés helyzetben vagyunk más népekhez képest e tekintetben. Például a román nyelvben öt ékezetes karakter van. Ebből kettő (a „t alatt vessző” és az „s alatt vessző” nincs benne az ***ISO-8859-2*** kelet-európai kódlapban. Ehelyett elkezdték ezeket a betűket „t alatt farkinca” és „s alatt farkinca” formában írni, mert ezek részei az ***ISO-8859-2*** kódlapnak, de ez annyira kielégítő megoldás, mint amikor magyarul hullámos és kalapos ékezeteket használunk az ***ő*** és ***ű*** betűkön. Az ***ISO-8859-16*** kódlap tartalmaz minden román ékezetes betűt, de ennek a kódlapnak az elterjedtsége nagyon csekély. A problémát súlyosbítja, hogy nincs egységes román billentyűzet-kiosztás. Több szabvány létezik, de egyik sem igazán jó (logikátlan helyen vannak a billentyűk), és ***Romániában*** a számítógépeket szinte kizárólag angol kiosztású billentyűzetekkel adják el. Újabb gondot jelentenek a hibás betűkészletek. Vannak, amelyekben léteznek a megfelelő karakterek, de rossz helyen. Vannak, amelyek csak a farkincás változatot tartalmazzák a vesszős helyett. Igazán jó nem nagyon van (beleértve a ***Windows*** alatt elérhető betűkészleteket is). Az eredmény az lett, hogy a románok a számítógépes kommunikációban leszoktak az ékezetek használatáról. A webes tartalom nagy része is ékezet nélküli. Most már nehéz is lenne változtatni helyzeten, mert például a ***Google***-ban is mindenki ékezetmentesen keres.

A szerb és a horvát fordítók élete sem fenékig tejfel. A szerb cirill ábécé teljesen megfelel a szerb nyelv leírására. Van azonban két cirill betű, amelynek latin átirata nem létezik, csak a ***Unicode*** kódtáblában. Ez az lj és az nj betűk, amelyek nem keverendők össze az l+j és az n+j betűpárokkal. Olyasmik, mint a magyar ábécé több karakterből álló betűi, de velünk ellentétben, érdekes módon mind a szerbek, mind a horvátok ragaszkodnának ahhoz, hogy ezek a betű egy karakterrel leírhatók legyenek. Ma már hiába terjed a ***Unicode***, latin betűs szövegeik gyakorlatilag mind a rossz formátumban vannak. További nehézség majdnem minden exjugoszláv ország fordítói számára, hogy nincs rögzített helyesírás és nyelvtan, nem tudják, mihez kell igazodni. Nacionalista megfontolásokból mind a szerb, mind a horvát, mind a bosnyák nyelvészek próbálják a lehető legjobban kihangsúlyozni a lényegében azonos nyelvek közötti különbségeket. Évente állnak elő új nyelvtani és helyesírási szabályokkal. A ***Magyarországon*** megszokott stabilitás – 1984 óta változatlan szabályok, pár évente alig egy tucat megváltozott írásmódú, főként idegen eredetű szó – számukra még csak álom. Egyes szláv nyelveknél gondot okoz a nyelvtani nemek és nyelvtani esetek használata a programok fordításakor. Amikor a fordítandó szövegben helykitöltő karakterlánc szerepel, amelynek helyére futási időben kerül valami, a feladat sokszor megoldhatatlan, vagy csak csúnya megoldás születhet, hiszen nem feltétlenül ismert előre a helykitöltő helyére kerülő szöveg neme, és a toldalék is függ a szó alakjától. Hasonló gond a magyarban is van, ráadásul nálunk a kétféle határozott névelő (a, az) alkalmazása az ismeretlen szöveg előtt is gondot jelent. Az egyik szerb előadó bemutatott olyan példákat, amelyek helyes lefordítása szerbre egyszerűen lehetetlen. Magyarban is találkozni néha nyakatekert szerkezetekkel hasonló okok miatt.

A körültekintő nyelvhasználat egyre lényegesebb, hiszen egy több nem hozzáértő ember használ számítógépet és szabad szoftvereket. Vicces helyzetek is adódhatnak abból, hogy ma már a számítógép-használat nem a szakemberek kiváltsága. Egyszer egy szerb ortodox pap megbotránkozva nézett, amikor azt tanácsolták neki, hogy kattintson az egérrel az ikonra, ő teljesen mást értett ez alatt.

Zárszavában ***Vedran Vučić*** hasznosnak értékelte a konferenciát. Többek között hangsúlyozta az együttműködés fontosságát, ami eszközök, módszerek, illetve rokon nyelvek esetén fordítások cseréjét jelenti. Szorgalmazta, hogy nagyobb hangsúlyt kapjon a teljesen kezdő számítógép-felhasználóknak szóló dokumentációk készítése, illetve fordítása, valamint az oktatóprogramok fordítása. Fontos feladatként említette az egymás országaiban élő kisebbségek megismertetése a saját anyanyelvükön működő szabad szoftverekkel, és ebben az anyaország aktivistáinak kell szerepet vállalniuk. Ennek folyományaként szervezés alatt áll az ***FSF.hu Alapítvány*** előadássorozata a ***Vajdaságban***, melyre a tervek szerint kora ősszel kerülhet sor.

Tímár András

Bemutatjuk a SUSE LINUX Professional 9.3-at

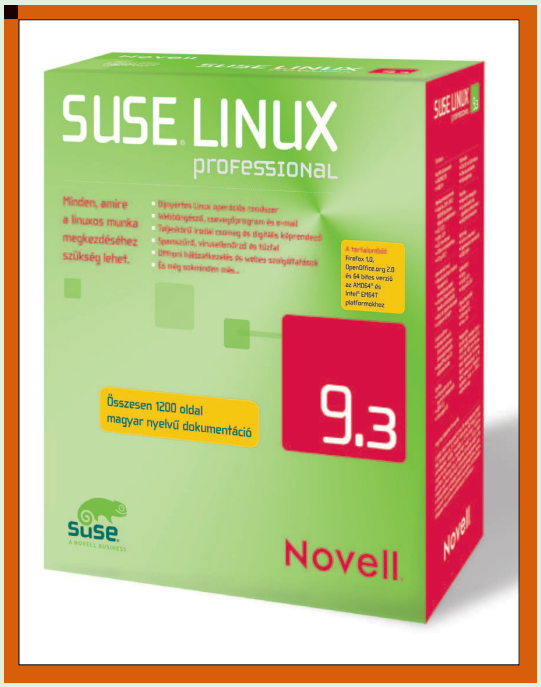
A Novell/SUSE néhány hónapja megjelent termékét bátran nevezhetjük az eddig legteljesebb Linux terjesztésnek. Ráadásul nemcsak a rendszer beszél magyarul, hanem a hozzá tartozó tekintélyes mennyiségű dokumentáció magyar változata is elkészült.

Áttekintés

A teljesség legnagyobb előnye természetesen a sokrétű felhasználhatóság. A ***SUSE LINUX Professional 9.3*** egyaránt használható asztali számítógépeken, otthoni, vagy kisvállalati kiszolgálókon, de használhatjuk nyílt forrású multimédia központként vagy programfejlesztésre is.
A legnagyobb meglepetés azonban az, hogy a sokrétűséggel nem jár együtt a bonyolultság. A rendszert a kezdők és a haladó vagy profi

felhasználók ugyanolyan egyszerűen telepíthetik és kezdhetik el használni. Aki előző tapasztalatai alapján pontosan tudja, hogy mit vár illetve várhat a rendszertől az a telepítés során azonnal testreszabhatja azt.
Aki pedig még csak most ismerkedik a Linux világával az választhat a különböző előre definiált telepítési módok közül.
Ez utóbbiak segítségével a teljes telepítési folyamat és a rendszer üzembe helyezése csupán néhány kattintást, illetve bizonyos alapvető információk megadását (névkiszolgáló címe, a rendszergazda jelszava) igényli.

A gép kiépítettségét a telepítő automatikusan felméri, és telepíti a szükséges összetevőket. Így hangkártyánk, hálózati kártyánk, ***DVD*** írónk, nyomtatónk jó eséllyel „magától” fog működni már az első bejelentkezés után.
Ami pedig a szoftverkínálatot illeti, 3000 nyílt forráskódú alkalmazás közül válogathatunk.
A ***Linux*** világában ma már az „alapfelszerelés” részének tekinthető ***OpenOffice.org*** irodai csomag, ***Firefox*** böngésző, ***Gimp*** képszerkesztő mellett a ***SUSE LINUX Professional 9.3***-ban helyet kaptak az olyan élvonalbeli, és még fejlődő technológiák is mint a ***Beagle*** vagy a ***XEN***. Előbbi egy keresőmotor míg utóbbi egyfajta gép a gépben funkciót (virtualizációt)valósít meg.
Linuxról lévén szó, a rendszer egyszerűen természetéből adódóan biztonságos, vagyis gyakorlatilag immúnis az e-mailekből és a webről származó vírusokkal és férgekkel szemben. Ezen túl természetesen számos a hálózati biztonságot szolgáló alkalmazást is tartalmaz.
Ami a dokumentációt illeti A ***SUSE LINUX Professional 9.3*** magyar nyelvű, 1200 oldalas részletes leírást



tartalmaz. Kapunk egy ***Felhasználói kézikönyvet***, amelyben alapvető információkat találhatunk az alkalmazásokról. Ez a könyvecske azt a célt szolgálja, hogy segítségével gyorsan elkezdhessük a rendszer használatát. A ***Rendszergazdai kézikönyv*** ugyanakkor átfogó információkkal szolgál a rendszer karbantartásával és üzemeltetésével kapcsolatban, vagyis elsősorban a rendszergazdáknak szól.
A vételár emellett 90 napos telepítési támogatást is magában foglal, vagyis ha gondunk akadna a ***Novell*** munkatársai rendelkezésünkre állnak.

INFOmarket-INFOtrend 2005
8 óra vásár, 8 óra pihenés, 8 óra szórakozás

Minden eddiginél több játék várja kilenc napon át a 4. Információtechnológiai és Telekommunikációs Vásár látogatóit. Az idei INFOmarket-INFOtrend az informatikai újdonságok mellett számítógépes bajnokságok színtere is lesz: a virtuális vadászpilóták, digitális focisták, és vérbeli Counter-strike-osok otthon érezhetik magukat szeptember 17. és 25. között a HUNGEXPO Budapesti Vásárközpontban.

Szárnyalás

A vadászpilóták légiharca a legmodernebb gépeken zajlik a „B" pavilonban. A kihelyezett ***Lock On*** és ***IL-2 Sturmovik*** szimulátoros játékok valósághű égi környezetet teremtenek. Az ***INFOmarket-INFOtrend*** közönsége idén először kísérheti figyelemmel a ***III. Magyar Repülőszimulátor Bajnokság*** előselejtezőit szeptember 17-19. között, illetve a döntőt 23-25. között. Az érdeklődők a játékról további információkat találhatnak a ➲ *www.jetfly.hu* honlapon, a kiállításra ellátogatók pedig hétfőtől péntekig ki is próbálhatják magukat vadászpilótaként.

World Cyber Games – FIFA 2005 Bajnokság

A szeptember 17-i hétvégén mindenki szurkolhat a ***World Cyber Games – Counter-strike offline*** selejtezők versenyzőinek. A játékról bővebb tájékoztatást találhatunk a ➲ *http://hu.worldcybergames.com* weboldalon. A 23-i hétvége a focirajongóké: cselezhetnek a ***FIFA 2005*** játékban és részesei lehetnek a ***FIFA 2005 Bajnokságnak***. Nevezni a ➲ *www.fifahungary.com* honlapon lehet.

Gépcsodák a kiállításon

A ***4. Magyar Tuning és Modding Kiállítás és Verseny*** először kerül megrendezésre az ***INFOmarket-INFOtrend***-en. A számítógép-designerek egyedi építésű számítógépekkel várják a nagyközönséget. A különleges gépek mellett a kiállításra elhozza újdonságait többek között ***Magyarország*** legnagyobb konzolforgalmazója. A felhasználók megismerhetik a legmodernebb vírusirtó programokat, ügyviteli- és nyelvoktató szoftvereket. A kiállítók irodai szoftverei közül több linuxos környezetben is működik. A kínálatban szerepelnek továbbá a hagyományos hardvertermékek, a legújabb ***MP3*** lejátszók, valamint vetítéstechnikai berendezések.

8 óra vásár, 8 óra pihenés, 8 óra szórakozás

A szeptember 17-25. között ***INFOmarket-INFOtrend*** napi 8 óra kikapcsolódást jelent minden korosztálynak, családoknak és fiataloknak egyaránt reggel 10-től, este 6-ig. A látogatókat a játék- és az informatika világába varázsolja, a párhuzamosan zajló ***BNV***, pedig még több szórakoztató programot kínál a nagyközönség számára a ***HUNGEXPO Budapesti Vásárközpontban***.

További információ:
➲ *www.infomarket.hu*

Érett-e a szabad szoftver? – Informatika érettségi buktatókkal

Az idei évvel teljesen megváltozott az informatika érettségi súlya: egészen idáig azok választották az informatikát, akik könnyen szerették volna letudni a kötelezően választott tárgyat, mivel a helyi követelményeknek egy kis felkészüléssel játszva meg lehetett felelni.

Idén viszont az informatika szakokra igyekvők a fizikát kikerülhették informatikával a felvételin. Most az emelt szintű vizsga feladatait fogjuk megnézni, mert alapvető eltérés nincsen. A szóbeli vizsga természetéből adódóan független a szoftverektől, leszámítva a konkrétan ezekre vonatkozó tételeket, így a gyakorlati vizsga kerül terítékre.

Buktató?

A nagy kérdés: érdemes-e szabad szoftverekkel nekivágni az érettségineknek vagy sem? Akinek kezdettől fogva követnie kellett a témával kapcsolatos fejleményeit, az tudja, hogy eleinte szép számmal akadtak erősen szoftverfüggő, hovatovább hibás mintafeladatok. Ennek egyik mintapéldája volt a képernyővédő készítés, melynek nehézsége attól függ, milyen grafikus lehetőségekkel rendelkezik a választott programnyelv.

Előny!

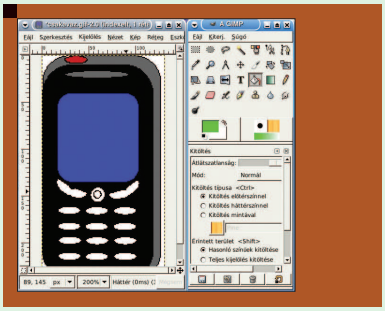
Természetesen ilyen is akad, hiszen ha nem volna, az egész cikk okafogyott lenne! Az informatika érettségi utáni napokban jelent meg több internetes lapban is, hogy egyes iskolákba nem érkezett lincensz, lejárt próbaverziók és ehhez hasonló akadályok nehezítették a diákok munkáját. Ehhez képest jó előre feltelepíthető a kért verzió a számítógépekre, nem kell minisztériumi licenszre várni, ha nem kereskedelmi terméket választunk. Másrészről a diák otthon is mindenből pontosan ugyanazt és ugyanúgy tudja használni, mint az érettségin, anélkül, hogy ez anyagi terhet jelentene: a választható szoftverek listája idén tartalmazta az ***Adobe Photoshopot***, ennek oktatási verziója egy átlagos számítógép árának akár felét is kiteszi! Ellenben a lista tartalmazta a ***Gimpet*** is, mint a legkomolyabb szabad szoftveres alternatívát. Vajon minden vizsgafeladathoz találunk alkalmas alternatívát?

Weblapszerkesztés

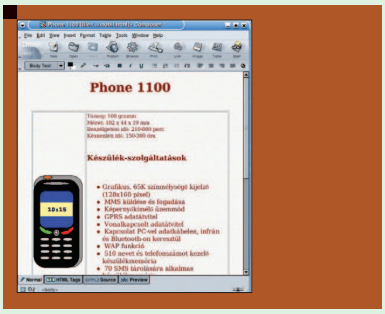
Mivel a web kimondottan platformfüggetlen, így aztán nem fordulhat elő, hogy egy adott weboldal csak ***Windows*** alatt készíthető el. A választék bőséges: ***Mozilla Composerrel és Nvu***-val is jól megoldhatók a kitűzött feladatok. Tudásszinttől függetlenül azonban érdemes Azt kapod amit látsz stílusban dolgozni, a szerkesztőprogramokat ebben a módban használjuk. Az idei feladat egy kis képszerkesztéssel kezdődött, egy kész képet kellett kiszínezni. A ***Gimppel*** a feladat könnyűszerrel elvégezhető. A weblapon sok teendő nincs, a kész szöveget kell táblázatba rendezni, felsorolásba rakni és formázni. A kész weboldalt bármelyik webszerkesztőben is nyitjuk meg, hibátlanul kezeli, éppúgy elkészíteni is tetszőleges programmal lehet. A kész munkát a ***Mozilla Composerben*** láthatjuk.

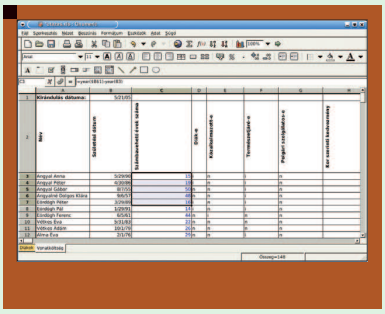
Táblázatkezelés

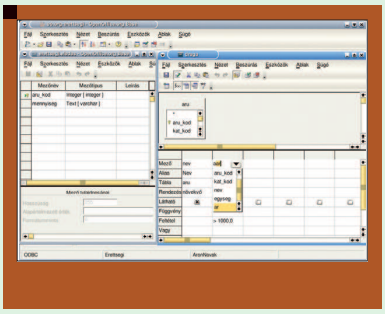
Ebben a feladatban szoftveroldalról a legfontosabb igény, hogy tudjon tabulátorokkal elválasztott adatokat

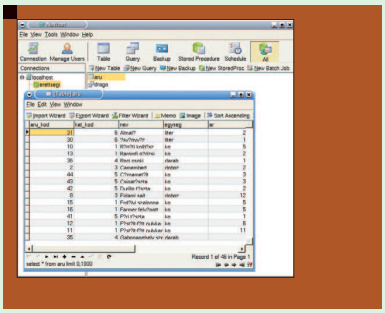


beolvasni egy szövegfájlból, mivel a mintapéldák és az érettségi feladata is így szolgáltatta a forrásadatokat. A ***Kspread*** már itt elvérzik, mivel nem lehet beállítani az importált txt fájl karakterkódolását, de más okból sems alkalmas: a kész feladatot sem tudja importálni, több helyen képtelen a képletek kezelésére. Az ***OpenOffice.org***-gal ellenben minden részfeladatot meg lehet oldani és az ékezethelyes importálás sem jelent gondot. A teszt során az 1.9-es bétaverziót használtam, remélhetőleg a következő érettségi idejéig stabillá válik (***http://development.openoffice.org/releases/OpenOffice_org_2_x.html***).
A mostani feladatban egy természetjáró csoport adatait kellett kezelni, szükség volt összetettebb függvényekre, mint például a ***VLOOKUP***-ra és szűrni is kellett az adatokat új munkalapra. Az ***OpenOffice.org***-ban semmi nehézség nem merült fel a ***Microsoft Excelhez*** képest, sőt egy valamiben kényelmesebb is volt: az ***OpenOffice.org*** bármely nyelvi verzióját is használjuk, nem kell újra megtanulni









a függvények neveit, mely a magyar Excelnél komoly fejtörést okozott néha. Viszont az *OpenOffice.org* viszonylag lomha és egy érettségi feladatmegoldásnál az idő sürget. Így szerencsére akad még egy alternatíva: a *Gnumeric* tökéletesen importálja mind a *txt* fájlt, mind az *Excelben* elkészített megoldást, minden szükséges eszközzel rendelkezik, villámgyorsan indul és a memóriafogyasztása pedig töredéke az *OpenOffice.org*-nak. Aki már megszokott egy linuxos táblázatkezelőt, nyilván maradjon a jól beváltnál, de különben a *Gnumeric* teljesen alkalmas a feladatra.

Adatbázis-kezelés

Maga az adatbázis-motor nem kérdéses, a *MySQL* vagy a *Postgresql* alkalmas a követelmények teljesítésére, már ami a matúrát illeti, de megfelelő felhasználói felületet, ahol *Access* szintű kényelmet és gyorsaságot kapunk, nem olyan egyszerű találni. Egy kereskedelmi programmal, a *Navicattel* minden felmerülő részfeladat gyorsan megoldható, van kényelmes lekérdezéskészítője, importálni is tud *txt*-ből, egyedül az ékezetes karaktereket nem kezelte le az alapbeállításokkal. A szoftver nem kerül többe, mint egy jó játékprogram (55 dollár az oktatási licenc), de mégis tovább kerestem, hogy van-e szabad szoftveres lehetőség. Igen, az előbbiekben a sebességéért elmarasztalt *OpenOffice.org 1.9*-es verziójában az *ODBC* kapcsolaton keresztül remekül el lehet érni bármilyen adatbázismotort és nagyon okos tündérei vannak, melyekkel az egyszerűbb lekérdezések könnyűszerrel összeállíthatók. Ráadásul még az ékezeteket is hibátlanul kezeli! Igaz ugyan, hogy kimondottan importálni nem tud, azonban a *txt* fájlból is képes létrehozni egy adatbázist és onnan a *MySQL*-ből létrehozott adatbázisba minden gond nélkül a másol/beilleszt parancsokkal átvihetőek a táblák és az adatok. Ügyes tündérek segítik az érettségizőt, valamint aki már megszokta az irodai programcsomag kezelőfelületét, annak barátságos lesz.
Az idei feladat főbb vonalakban: három tábla importálása, aztán sokféle több táblás lekérdezés gyártása. Igazándiból egyik szoftvert sem hozza zavarba.

Programozás

Aki +7 felvételi pontért vállalja az emelt szintet, annak kell tudni elemi algoritmusokat programozni. Ezen a területen talán előnyben van a szabad szoftver, mivel sok kiforrott *IDE* érhető el a különböző programnyelvekhez. Az *Eclipse*, a *Kdevelop*, az *Anjuta* kényelmessé teszi a munkát. Idén egy szövegfájlból beolvasott adatsoron kellett műveleteket végezni: rendezni, keresni és összegezni. Valószínűleg egy szkriptnyelvvel lehet a legkényelmesebben megoldani a feladatot, mondjuk a *Pythonnal*. A szövegszerkesztő illetve a fejlesztési környezet ez esetben végképp ízlés dolga, bármilyen programnyelvben megoldhatóak a feladatok. Akár *gcc* és *vi* segítségével is.

Összegzés

Az összes említett program megtalálható a nagyobb *Linux* terjesztések mindegyikében, egyedül a *Navicat* nem. Ennek 30 napos tesztverziója a *www.navicat.com* címen érhető el.

Amennyiben a magyar *OpenOffice.org*-ból az 1.9-est szeretnénk használni, hiszen az itt bemutatott adatbázis-kezelő abban érhető el, az a *http://ftp.linux.cz/pub/localization/OpenOffice.org/devel/680/* címről tudjuk letölteni.
Szerencsére nincs olyan részfeladat, amit sokkal nehezebben, lassabban lehetne megoldani szabad szoftverekkel. A javasolt szoftverek nagy része egyébként *Windows* alá is elérhető. A feladatsoron áll vagy bukik, hogy általánosake, idén a megoldandó feladatok ilyen téren kifogástalanok voltak. A szövegszerkesztés kimaradt az emelt szintű feladatsorból a májusi vizsgán, helyette volt a weblapszerkesztés, de e tekintetben sem kell szégyenkeznie a linuxos irodai szoftverkörnyezetnek. Egyfelől tehát bátran bele lehet vágni tisztán szabad szoftveres környezettel az érettségineknek, másfelől, hiszen az új érettségi rendszer minduntalan a gyakorlatias tudást hangsúlyozza, ismét kiderült, hogy az átlagos munkahelyi felhasználást már régóta képes a szabad szoftver világa magas szinten kiszolgálni.

Novák Áron
(aaron@szentimre.hu)
BME-VIK-es gólya, műkedvelő rendszergazda. Jelenleg leginkább a NetBeans-szel és mindenféle hordozható eszközzel foglalkozik, legalábbis mindazokkal amelyeket meg lehet szólaltatni Linux alatt.

KAPCSOLÓDÓ CÍMEK

Az idei emelt szintű feladatsor PDF formátumban:
➲ http://www.om.hu/letolt/okev/doc/erettsegi_2005/e_info_fl.pdf

Novell és SUSE – Út a csúcsra

A Linux kezdetben nem volt több, mint érdekes próbálkozás arra, hogy értelmes célra használják a 386-os processzorok képességeit. Ebben az időben senki, valószínűleg maga Linus Torvalds se gondolta komolyan, hogy ebből egyszer olyan termék lesz, amire világcégek teljes üzleti stratégiákat építenek. Aztán jött a meglepetés...

A nyílt forrás mint üzlet

A szakma addigi nagyjai számára az első hidegzuhany a ***Red Hat*** volt, olyannyira, hogy az ***Intel*** főembere állítólag még káromkodott is egy nagyot, amikor arra a kérdésére hogy ***„Mi az ördög az a Linux?”*** a tanácsadói közül senki nem tudott válaszolni.

A ***Linux*** azonban még sokáig barátságtalan és rejtélyes rendszer maradt, üzleti szempontból pedig a legkomolyabb és – különösen a konkurencia által – leggyakrabban hangoztatott vád vele kapcsolatban az volt, hogy szemben a kereskedelmi termékekkel ***„nem áll mögötte senki”***.

És lássuk be, ez valahol igaz is volt. Míg egy otthoni „műkedvelő” felhasználó ráér azzal tölteni az idejét, hogy különböző hibajavítások után kutat az internet sötét bugyraiban, ha valami történetesen nem működik a gépén, ugyanez egy ezer gépet üzemeltető vállalatnak adott esetben óriási anyagi kárt okozhat.

Amint azonban egyre több komoly cég ismerte fel a ***Linuxban*** rejlő lehetőséget, körvonalazódni kezdett egy nyílt forrású üzleti stratégia. Ennek lényege, hogy szemben a szoftvervilág eddigi működési mechanizmusával itt az igazi értéket nem maga a szoftver, hanem a hozzá kapcsolódó szolgáltatások jelentik. A szoftver ingyenes, bárki letöltheti, és használhatja, amire csak akarja. Az igényesebbek vásárolhatnak ugyan dobozos változatot, de tulajdonképpen ez se kerül többe, mint a csomagban található dokumentáció.

Aki viszont „nagyban játszik”, biztonságban akarja tudni vállalkozása informatikai infrastruktúráját, s ezért azt szeretné, hogy mindig legyen egy csapat, amely a segítségére siet ha baj van, immár megvásárolhatja ezt a szolgáltatást. A szoftver, amit kap, lényegében ekkor sem különbözik attól, mint amit bárki szabadon letölthet, viszont immár senki sem tagadhatja az eddig hiányolt támogatás meglétét.

Netware + SUSE = Novell Open Enterprise Server

A ***SUSE*** felvásárlásával nemrég a ***Novell*** is belépett a nyílt forrású szoftverek piacára. A röviddel ezután piacra dobott termék, a ***Novell Open Enterprise Server*** a két cég legfontosabb termékeinek, vagyis a ***NetWare-nek*** és a ***SUSE LINUXnak*** az összeolvadásából keletkezett.

A cél egyértelműen a nyílt forrású váltás megkönnyítése volt, hiszen a tapasztalatok azt mutatják, hogy ez az a dolog, amitől egy vállalat – az adóellenőrök után – a legjobban fél.

A ***Novell Open Enterprise Server*** felhasználói tulajdonképpen mindkét terméket megkapják egy olyan integrált csomagban, amely lehetővé teszi, hogy a rendszergazdák a telepítés vagy a tesztüzem során eldönthessék, hogy melyik kernelt illetve szolgáltatáscsomagot kívánják telepíteni. Akármelyik mellett döntenek is azonban, a részrendszerek közötti együttműködés a későbbiekben garantáltan zökkenőmentes lesz. Így elképzelhető, hogy míg

A kiválasztott

Az idei őszön a természet és a magazin megújulása mellett Linux-logót is (be)újítunk.

A Linuxvilág szerkesztősége nem éri be az 1996-os, Larry Ewing egere-szülte pingvin grafikamásolatával, hanem hús-vér-toll úszómadarat választ magának és olvasóinak. Az 57-es szám megfelelő érettséget sugall, hogy felnőve a nevelői feladathoz, örökbefogadjunk egy kissé molett testalkatú, totyogó pingvin-példányt a Budapesti Állatkert gyűjteményéből.
Ehhez kérjük olvasóink segítségét: szavazzanak a számukra legszimpatikusabb csőrösre a www.linuxvilág.hu weboldalon!
A kiválasztottról – mindennapjairól, szokásairól, linuxos kapcsolatairól cikksorozatban tudósítunk.

A gyermekátadási ceremóniára a Budapesti Állatkert Pálma Cukrászdájában (1146 Budapest, Állatkerti krt. 6–12.) kerül sor 2005. szeptember 14-én (szerdán), 15.30 órakor, mely sajtótájékoztatóval, csemegézéssel egybekötött ünnepségre ezúton szeretettel meghívunk minden kedves érdeklődőt! Részvételi szándékukat kérjük jelezzék a 477-0443-as telefonszámon, illetve a *hlena@kiskapu.hu* levélcímen.

a pénzügy jelenlegi alkalmazásai inkább a *NetWare*-t igénylik, addig a marketing részleg már linuxos alapokon folytatja a munkáját. A dogozók, az alkalmazások és az üzleti folyamatok számára ez a műszaki különbség észrevétlen marad.

Az üzleti világ reakciói

Egy anekdota szerint ***Liszt Ferenctől*** egyszer valaki megkérdezte, nehéz-e zongorázni, mire a mester ennyit válaszolt: ***„Egyáltalán nem nehéz, csak a megfelelő időben, a megfelelő erővel a megfelelő billentyűt kell leütni. Ennyi az egész.”***
2005-ben a ***Novell Open Enterprise Server***, mint a világ első vegyes forráskódú szoftvere nyerte el a ***Network Magazine Innovációs Díját***. A lap ***„áttörést jelentő”*** technológiaként jellemezte a terméket mondván ***„az innováció időnként nem több, mint a megfelelő termékbe a megfelelő képességeket csomagolni, és azt megfelelő áron árusítani”***.
Áprilisban a ***Deutsche Bahn, Németország*** nemzeti vasúttársasága bejelentette: üzletviteléhez szerverplatformként a ***Novell SUSE LINUX Enterprise Servert*** választotta, s további IT stratégiáját erre alapozza. A ***DB Systems***, a ***Deutsche Bahn*** informatikai szolgáltatója több mint 300 ***Intel*** alapú szervert használ, amelyeken kivétel nélkül ***SUSE LINUX*** üzemel majd.
Tekintettel a rendszer méretére a teljes átállást a cég több lépésben hajtja végre. A ***Lotus Notes*** termékhez hasonló alkalmazások már átkerültek ***IBM*** platformra, és szintén a ***SUSE LINUX Enterprise Serveren*** futnak.
A következő lépés az ***SAP*** szolgáltatások átállítása lesz. Az utolsó szakasz várhatóan 2005 végére valósul meg. Ekkor a ***DB Systems*** webkiszolgálóit és a hálózati infrastruktúra egyéb fennmaradó szervereit is – többek között az adatbázis és alkalmazásszervereket – átállítják.

A jövő

Ha egy nagyvállalat IT vezetőjének olyan horderejű kérdésben kell döntést hoznia, mint az elavult infrastruktúra cseréje, nincs könnyű helyzetben. Míg az eladók célja értelemszerűen saját portékájuk magasztalása, neki nyilvánvalóan objektív adatokra van szüksége ahhoz, hogy eldönthesse, melyik cég termékeire célszerű alapoznia az új rendszert.
Júniusban a ***Novell*** bejelentette az első olyan üzleti megoldást, amely tagja a cég ***Hitelesített Konfiguráció Programjának (Validated Configuration Program)***. A csomag ***HP*** szervereken futó ***Novell***, ***JBoss*** és ***Oracle*** szoftverekből épül fel.
A hitelesítési program célja kettős: egyrészt meghatározza, hogy ***Linux*** alatt mely célokra milyen szoftverek használhatók, másrészt az üzleti élet számára fontos szempontok (használhatóság, platformfüggetlen, integrálhatóság) szerint minősíti ezeket.
A kezdeményezés újszerűségét az adja, hogy az eddigi tanúsítási eljárásokkal szemben itt nem csak az egyes alkotóelemeket vizsgálják meg, hanem a teljes konfigurációt. Ez a minőségi megközelítés nyilván sokkal jobban idomul a vállalati adatközpontokban használatos összetett konfigurációkhoz, hiszen átfogóbb, és a beruházási döntések szempontjából sokkal használhatóbb adatokat szolgáltat azokról.

2005 linuxos erőműve

Egyesek nagy teljesítményű, kétprocesszoros rendszert akartak építeni. Mások csendes gépre vágytak, ami zenei célokra is megfelel. Úgy döntöttünk, mindkét igényt kielégítjük.

A 2005-ös évet nyugodtan hívhatjuk az energiakezelés évének. A processzorgyártók számtalan tanulmányban taglalják a wattokkal való takarékoskodást, és a februárban megrendezett ***LinuxWorld*** konferencián és kiállításon is komoly figyelmet kapott az energiakezelés.

Talán az iparág aggódni kezdett a globális felmelegedés miatt? Talán az informatikai cégek vezetői több kardhalat akarnak enni, és csökkenteni akarják a gyárak higanykibocsátását? Aligha. Napjaink kiszolgálóiban egyre több és forróbb processzor üzemel, és a felhasználók légkondicionáló rendszerei képtelenek megbirkózni ekkora terheléssel. A ***NASA***-nak vízhűtéssel kellett ellátnia 10240 processzort tartalmazó, ***Columbia*** nevű fürtjét, ahogy arról a januári számban írtunk is.

Minden megtakarított watt 3,6 kJ-nyi, illetve, hogy pontosak legyünk, 3,4 BTU-nyi hőt jelent, amivel könnyebbé válik a felhasználó dolga. A penge kiszolgálókkal és az akár négy processzort is tartalmazó egy egység magas rendszerekkel teli adatközpontokban mindez a hő összeadódik.

Az asztali ***Linuxok*** mohón falják fel a linuxos kiszolgálók sok milliárd dolláros piacának maradékait, így az energiafogyasztás az asztali gépeknél is egyre nagyobb figyelmet kap.

A ventilátorok hangosak. Ha a processzorokat jobb energiakezeléssel látjuk el, akkor kevesebb hőt termelnek, és kevesebb ventilátorra lesz szükségünk, illetve a meglévőket alacsonyabb fordulatszámon tudjuk működtetni. Mi ventilátorok helyett mást választottunk, mint még lesz róla szó.

Mi sem természetesebb, hogy az energiafogyasztás a hordozható számítógépek és az egyéb mobil eszközök esetében is fontos. A hálózati tápellátás nélküli üzemidő növelésének kérdésével későbbi írásainkban fogunk foglalkozni.

Alaplap: a rendszer szíve

Kedveljük a ***Tyan*** alaplapjait, ahogy az egyedi linuxos rendszereket építő cégek is. A négy darab ***Opteron*** processzor befogadására képes ***Tyan Thunder K8QS Pro*** kicsit későn jött ki ahhoz, hogy a múlt év linuxos erőművének részévé válhasson. Az ***AMD 8000***-es sorozatú lapkakészletére épül. Amikor lapkakészletet mondunk, egy picit más hardverelemre gondolunk, mint amit az ***Intel*** alapú gépekben láthatunk. Az ***AMD64*** alapú rendszerekben minden processzor saját, beépített memóriavezérlővel és saját memóriabankkal rendelkezik, ezek közt ***HyperTranspor*** kapcsolat áll fenn. Az ***AMD64***-es, többprocesszoros gépek valójában kisebb ***NUMA (Non-Uniform Memory Architecture, nem egységes memóriaarchitektúra)*** gépek, és ezeknél magában a lapkakészletben nincs is memóriavezérlő.

Tavaly egy ***Celestica A8440***-es szekrénybe szerelhető váz szolgált a linuxos erőmű alapjául. Bár az előre szerelt házakkal és tápegységekkel sok időt megtakaríthatunk, a tavalyi gép meglehetősen hangosra sikeredett. Idén – a szokásos megoldáshoz visszatérve – minden alkatrészt külön-külön válogattunk össze.

A ***K8QS Pro*** két ***PCI-X*** busszal rendelkezik, az A és a B jelűvel. A B busz két 133 MHz-es ***PCI-X*** foglalat számára van elkülönítve, az A pedig kettő darab 66 MHz-es ***PCI-X*** és egy darab hagyományos ***PCI*** foglalat kiszolgálásáért felelős. A hálózati kapcsolatok létrehozását kettő darab ***Broadcom BCM5704C Gigabit Ethernet*** csatoló segíti, ezek szintén az A buszra csatlakoznak.

A megszokott kapuk is a rendelkezésünkre állnak, ezek közül mi csak az ***USB*** kapukat vettük igénybe.

A ***SCSI*** és az ***ATA*** vezérlő kiegészítő jelleggel kérhető, erre nem árt odafigyelni, ha linuxos erőművünk építése közben az alaplapot hagyományos kiszolgáló szerepet játszó gépbe akarjuk beépíteni.

Ebbe a kiváló alaplapba az elérhető ***Opteron*** processzorok közül a legjobbakat (***846 HE*** típusjelzés, 2 GHz-es órajel, 1 MB másodszintű gyorsítótár) helyeztük be, azokból is mindjárt négyet. A rendszer tesztelése közben elérhetővé vált újdonságokat a széljegyzet taglalja. A rendszerbe a lehető legtöbb memóriát, 32 GB-os építettünk.

A házrajongók szerencsétlenségére az alaplap ***SSI MEB*** formátumú, vagyis mérete 13” x 16”, azaz 330,2 x 406,4 mm. Számunkra ez nem okozott gondot, ugyanis idén egyedi házat használtunk, de a méret mindenképpen korlátozza azon házak körét, amelyek közül választhatunk.

1. kódrészlet: A /etc/fstab fájlban szereplő lemezrészek

```
LABEL=/nstor-OS       /          ext3    defaults          1 1
LABEL=/cfboot         /boot      ext3    defaults          1 2
LABEL=/nstor-DATA     /u1        ext2    defaults          1 2
none                  /dev/pts   devpts  gid=5,mode=620    0 0
none                  /dev/shm   tmpfs   defaults          0 0
none                  /proc      proc    defaults          0 0
none                  /sys       sysfs   defaults          0 0
```

1. ábra Mi van az irodádban, Justin? A hűtést egy nulláról induló rendszerrel, az lm_sensors segítségével teszteltük.

Amikor kiválasztjuk egy egyedileg épített rendszer házát, legyen szó akár a folyó év erőművéről, akár más gépről, mindig valamivel nagyobbat választunk, mint amit a neves gyártók egy hasonló géphez kínálnának.

A kisebb házakhoz kevesebb anyag kell, és szállítani is olcsóbb őket, mi viszont egyediségre törekszünk, ezért több helyre van szükségünk, egyrészt az eszközök hozzáadásához, másrészt a gépben végzett munkához.

Adattárolás

Ha teljesen csendes rendszert akarunk építeni, akkor az adattárolást a gép házán kívülre kell száműznünk. Régen ezt *NFS*-sel vagy három méteres kábelekkel csatlakozó külső *SCSI*-házak segítségével lehetett megoldani, ám azóta bővültek a lehetőségek.
Ha a meghajtók távol tartására *USB*, *FireWire*, *SCSI* (természetesen ez nem maradhat el), *Fibre Channel* vagy éppen *ATA over Ethernet* megoldásokat használhatunk, utóbbiról 2005. júniusában mi is írtunk. A különálló meghajtóház többé nem a vállalati adatközpontok kiváltsága.
További lehetőség a hálózatról végzett rendszerindítás, majd a tárolóhely *NFS*-en keresztüli befűzése. Mivel a *Penguin* vállalati kiszolgálószobákban látott eszközökkel dolgozik, és a *Fibre Channel* lenyűgöző eredményeket hozott a teljesítménytesztek során, végül mellette döntöttünk: egy *nStor 4320F Fibre Channel RAID*-házat választottunk, amelyben 18 GB-os *Hitachi* meghajtók tárolták az operációs rendszert, további, nagyobb méretű *Seagate* meghajtók pedig további tárhelyet biztosítottak.
Mivel önálló, a rendszerindítás tekintetében más kiszolgálótól nem függő rendszert akartunk összeállítani, a rendszerindítás céljára beszereltünk egy *256 MB-os Sandisk CompactFlash* kártyát. Ez a gép számára pontosan úgy látszik, mint bármely *ATA* meghajtó, vagyis bármilyen személyi számítógépes alaplap képes róla betölteni a rendszert.
USB-kulcs használatára is gondoltunk, ám ahhoz bele kellett volna nyúlni az initrd és a GRUB beállításaiba. Természetesen annak is vannak előnyei, ha a rendszerindításra használt eszközt ki lehet húzni a gépből, illetve el lehet különíteni, de nem számoltunk azzal, hogy repülőtereken keresztül fogjuk utaztatni a gépet, titkosított, bizalmas adatokkal teli meghajtókkal.
Ha csendesnek készülő linuxos gépünket mindig rajta akarjuk hagyni a hálózaton, akkor rugalmasabban dönthetünk a rendszerindításról, és például *PXE*-t is használhatunk.
Ha viszont alkalmanként valamelyik barátunkhoz is el akarjuk vinni a gépet, és ott szeretnénk zenéket lejátszani vele, akkor szükségünk lesz a független rendszerindítás lehetőségére.

A 2005-ös év linuxos erőművének alkatrészei

Alaplap: Tyan Thunder K8QS Pro (S4882)

Processzor: 4 db AMD 846HE Opteron

Memória: 8 db 4 GB-os Registered ECC Samsung DDR PC2700 CL 2,5 DIMM

Tápegység: 510 W-os, átépített PC Power and Cooling Turbo-Cool 510 ATX

Ház: egyedi, Matt Fulvio tervei alapján Trevor Sherard készítette

Fibre Channel: Qlogic 2342 2 Gb Fibre Channel csatoló, két kapuval, 133 MHz, PCI-X

Rendszerindító eszköz: 256 MB-os Sandisk CompactFlash kártya, DCFB-256-A10

Adattárolás: nStor 4320F Fibre Channel RAID-ház

Merevlemezek: 2 db 18 GB-os, 10000 fordulat/perces Hitachi DK32DJ-18FC Fibre Channel meghajtó, RAID 1 tömbben (operációs rendszer) és 6 db 73 GB -os, 10000 fordulat/perces Seagate ST373405FC Cheetah 73LP FC Fibre Channel meghajtó, RAID 10 tömbben

Grafikus kártya: PNY NVIDIA Quadro NVS 280 PCI

Kijelzők: 2 db ViewSonic VX2000 20"-os LCD monitor, 1600 x 1200-as felbontással

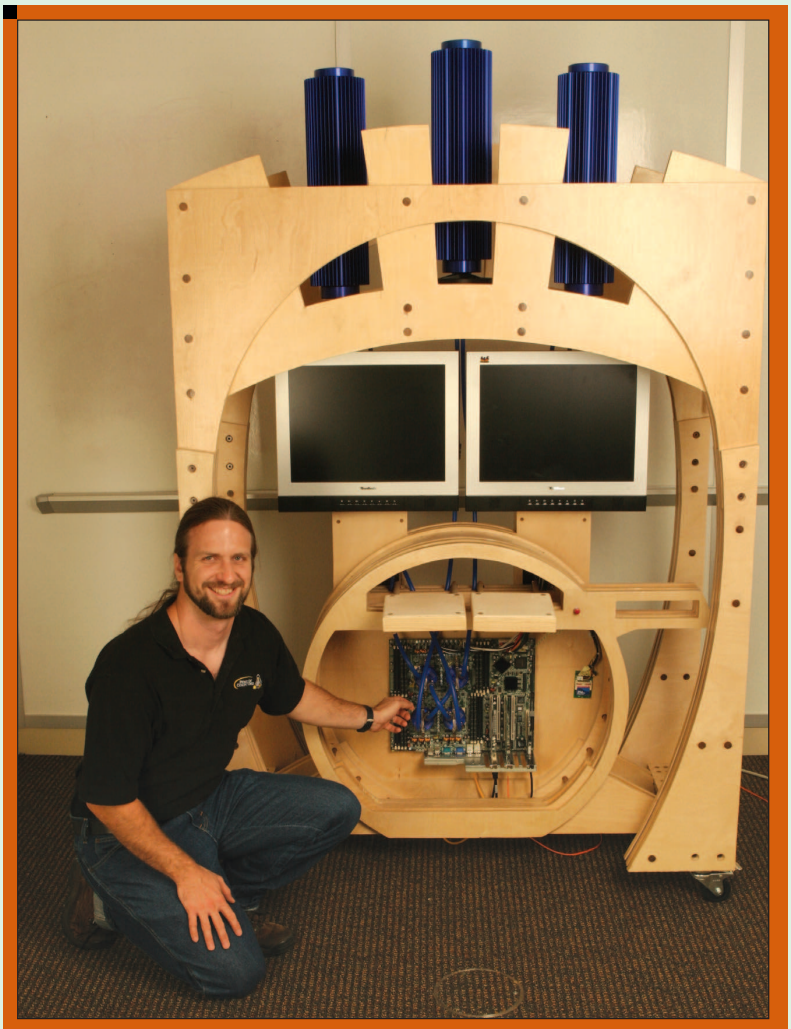
Hangkártya: RME HDSP9652 PCI hangkártya

Hang be- és kivitel: 36 csatornás, 24 bites, 96 kHz-es RME Multiface be- és kiviteli egység

Hűtőrendszer: 3 db Zalman Reserator 1s

Vízhűtőblokkok a processzorokon: 4 db Zalman ZM-WB2 Gold blokk

Tápegység hűtése: egyedi, tervezte és készítette a Global Precision



2. ábra Semmi sem szivárog! Mindenki látja?

A *Penguin* csapat úgy tervezte, hogy elviszi a gépet a *LinuxWorld* rendezvényre, márpedig, ha bemutatóra kell cipelni egy gépet, akkor csak jó, ha eggyel kevesebb dolgot kell üzembe helyezni a helyszínen.
Aki csendes *Linuxos* gépet épít és telepít, az akár kettős megoldást is kidolgozhat: *NFS*-t alkalmaz a felhasználók kezdőkönyvtárainak, a vállalati */usr/local/bin/* könyvtár és az egyéb olyan anyagok tárolására, amelyeknek ugyan naprakészeknek kell lenniük, de vészhelyzetben nélkülözhetők; míg a gép saját fájlrendszerét a nagyméretű munkafájlok – például a rendszer csúcskategóriájú hangkártyájától érkező adatok – tárolására tartja fenn.
Végül, hogy még az egérkattintások se törjék meg a csendet, a *Penguin* alapítója, *Sam Ockman* egy érintőpadként működő *TouchStream LP* billentyűzetet javasolt, melynek nincsenek mozgó részei. Egyben mutatóeszközként is szolgál, és a műveletekhez egérgesztusok hozzárendelését is lehetővé teszi.

Hangrendszer

Ez az első alkalom, hogy a linuxos erőműbe profi hangkártyát szereltünk. Egy csendes gép hol is találhatna jobb helyre, mint egy hangstúdióban?
Az általunk kiszemelt *RME Hammerfall HDSP9652* kártya akár 52 csatorna kezelésére is képes; beszereztünk hozzá egy *Multiface* nevű külső dobozkát is, amelyen 8 darab □"-os aljzat, valamint optikai, koaxiális és *MIDI* csatlakozó található.

A kártya gyakorlatilag kisebb stúdió egyetlen egységben, ugyanis belső keverővel rendelkezik, ami lehetővé teszi a jelek a kártyán belüli irányítását, minimális késleltetésekkel és a processzorra nézve elenyésző mértékű terheléssel; továbbá például a hagyományos, szalagos magnókat idéző szolgáltatásokkal is rendelkezik.
A legjobb benne mégis az, hogy az ***RME 2000*** óta támogatja az ***Advanced Linux Sound Architecture (ALSA) Projectet***, vagyis esetében a ***Linux*** felhasználók nem csupán másodrangú polgároknak számítanak. Az ***RME*** webhelye szerint ***„a Hammerfall ALSA-támogatása végre megtöri a mindannyiunk számára unalmassá vált, a tyúk vagy a tojás kérdéséhez hasonló nincs professzionális hardver vagy illesztőprogram – nincs professzionális hardver kört”***.
2003. októberi számunkban ***Peter Todd*** részletesen is tárgyalta a ***Hammerfall HDSP*** kártyák használatához szükséges eszközöket.
A megjelenítésre egy viszonylag egyszerű kártyát alkalmaztunk (lásd az internetes forrásokat). Hiába szerettük volna érdekes és újszerű megjelenítési képességekkel felruházni a gépet, az illesztőprogramok terén továbbra is problémákat tapasztaltunk (lásd a keretes részt).

Hőmérsékletkezelés

Hogyan tartatjuk megfelelő hőmérsékleten az eszközöket? A legfontosabb, hogy ne kezdjünk el játszadozni a különféle hardverkombinációkkal, amíg nem tudjuk, hogyan mérhetjük a módosítások által a rendszer hőmérsékletére gyakorolt hatást. Sőt, semmit ne változtassunk meg, amíg nem tudjuk, hogyan fogjuk mérni a változás hatását.
A jó hír az, hogy a processzorok és az alaplapok gyártói a legfontosabb alkatrészeket hőmérsékletérzékelőkkel is ellátják, ezek jeleit egy alapvető eszközzel, az `lm_sensors` segítségével tudjuk figyelni.
A meghajtók hőmérsékletét nem mértük, hiszen ezek külön házba kerültek, de a ***smartmontools*** (lásd a forrásokat) segítségével ezt is könnyen megtehetjük.
Rendeltünk néhány tételt a kiváló vízhűtőkészleteket kínáló ***Zalmantól***. A leglátványosabb darab a ***Reserator 1***, mely egy fél méter magas, 2,5 liter vizet tartalmazó, kombinált víztartály és hőleadó. A ***Reserator*** mellett a processzorokhoz is rendeltünk egy-egy vízblokkot, illetve a szükséges csöveket is beszereztük.
A hőmérsékleti becslések azt mutatták, hogy nem lesz szükségünk processzoronként egy-egy ***Reseratorra***, vagyis két-két processzorhoz egy-egy ***Reseratort*** illesztettünk, illetve a tápegység is kapott egyet.
A ***Reserator*** egy 5 wattos szivattyúval rendelkezik, ami sajnos nem zajtalan, ezért a ***Reseratorokat*** át kellett alakítanunk hőáramlásos működésűre. Alapkiépítésben a ***Reserator*** bemenete és kimenete közel vannak egymáshoz, ezért mindegyik ***Reseratorba*** beszereltünk egy a forró vizes bemenettől a tetejéig futó csövet.
Működött. A processzorhőmérséklet körülbelül 50° C-ig kúszott fel, és a processzorokat és a ***Reseratorokat*** összekötő csövek eléggé felmelegedtek ahhoz, hogy beinduljon a hőáramlás. Normál használat mellett a hőmérséklet 47-48° C körül maradt, teljes terhelés mellett pedig nem ment 50° C fölé.
A tápegység hűtése már keményebb feladat volt. A ***Zalman*** legerősebb ventilátor nélküli tápegysége 400 W-os, ám a négyutas alaplapnak ez kevés volt. Így a ***PC Power and Cooling Turbo-Cool 510 ATX*** egysége mellett döntöttünk.

A teljesítménytesztek eredményei

dbench 100 szimulált ügyféllel:

```
%dbench 100
Throughput 1234.57 MB/sec (NB=1543.21 MB/sec
➥12345.7 MBit/sec)
```

Bonnie++ 1.03 – egy pontosabb, a merevlemezek sebességét mérő program:

- Soros kivitel, karakterenként: 58577 Kb/s, 98 % CPU
- Soros kivitel, blokkonként: 281032 Kb/s, 50 % CPU
- Soros kivitel, újraírás: 52603 Kb/s, 18 % CPU
- Soros bevitel, karakterenként: 34717 Kb/s, 58 % CPU
- Soros bevitel, blokkonként: 90097 Kb/s, 11 % CPU
- Véletlenszerű léptetés: 257,5/másodperc
- Soros létrehozás: 5924 fájl/másodperc
- Véletlenszerű létrehozás: 6056 fájl/másodperc

Postmark teljesítményteszt

A ***Postmark*** egy leterhelt levelezőkiszolgáló működését szimulálja. 20000 alapfájllal és 100000 tranzakcióval az alábbi eredményeket kaptuk.

Idő:
- 46 másodperc (összesen), ebből 40 másodperc tranzakciókezelés (2500 db/másodperc).

Fájlok:
- 70128 létrehozás (1524 db/másodperc); Létrehozás önmagában: 20000 fájl (5000 db/másodperc);

Tranzakciókkal vegyesen:
- 50128 fájl (1253 db/másodperc)
- 49656 olvasás (1241 db/másodperc)
- 50199 hozzáfűzés (1254 db/másodperc)
- 70128 törlés (1524 db/másodperc)

Törlés önmagában:
- 20256 fájl (10128 db/másodperc);

Tranzakciókkal vegyesen:
- 49872 fájl (1246 db/másodperc)

Adatok:
- 303,46 MB beolvasva (6,6 MB/s)
- 436,18 MB kiírva (9,48 MB/s)

A rendszermag lefordítása:
- 50 másodperc

Felmerült a saját tápegység építésének ötlete is, de ezt elvetettük, mert fontos, hogy az összetevők megfelelő sorrendben kapják meg a tápfeszültséget, és a ***PC Power and Cooling*** már megoldotta ezt a kérdést helyettünk. A hűtés gondja viszont megmaradt.
Itt lépett be a képbe a lakatosmesterség. ***Phil*** elment a ***Global Precision*** nevű fémmegmunkáló műhelybe, ahonnan három munkafázist rendeltünk meg. Először levágták a tápegység hűtőbordáinak eredeti lamelláit, ezzel sima, a vízhűtés blokkjainak felszerelésére alkalmas felületek jöttek létre. Ez után elkészítették magukat a vízhűtés blokkjait; itt kék színű, eloxált alumíniumot választottunk, ez ugyanis illeszkedett a ***Zalman*** alkatrészekhez. Végül két Y-csatlakozót is kértünk, ezek feladata a vízáram a két hűtőblokk közötti elosztása lett.
A tápegységből eltávolítottuk a ventilátorvezérlést – többé úgysem volt rá szükségünk.

Ház

Egy ilyen gép befogadására, igényeinek kielégítésére kevés ház képes. Idén egyetlen választásunk volt: egyedi megoldást kidolgozni. Az idei ház akrilablakokat kapott, melyeken keresztül látható a hűtőrendszer, valamint beépített támasztékokkal rendelkezik a ***Reseratorok*** és az ***RME Multiface*** számára.

Összefoglalás

Mindezek után, ha nehéz is elfogadni, de a való életben a legtöbb számítógépben sem 52 csatornás hangrendszerre, sem ***Fibre Channel*** alapú adattároló alrendszerre nincs szükség. Azonban a szokatlan összeállítások azok, amelyek képesek segíteni az igazán kreatív tevékenységeket, és örömünkre szolgál, hogy a Linux semmiben nem lesz gátunkra, segítségével bármibe belefoghatunk.
Aki azzal indul, hogy mi a lehetséges, majd kiveszi a számára szükségtelen elemeket, az bízhat benne, a gépe meg fog felelni az igényeinek. Reméljük, hogy olvasóink bármilyen számítógép építése mellett döntsenek is, sikerült néhány ötletet meríteniük az idei év linuxos erőművéből.

A jövő hardvere és a múltba révedő ügyvédek

Ez mindig bejön. Azok az új termékek, amiket ki szeretnénk próbálni az év linuxos erőművében, mindig pontosan akkor jelennek meg, amikor már a munka közepén járunk.
A hőmérséklettesztek elvégzéséhez már túlságosan későn jelentek meg az ***AMD*** kétmagos ***Opteron*** processzorai, amelyekkel a meglévő, négy foglalatot tartalmazó alaplapunkat felhasználva is építhetünk nyolcutas rendszert – elég egy ***BIOS***-frissítést végrehajtanunk. Ma még tízezer dollár (körülbelül kétmillió forint) négy ilyen processzor, ám várakozásaink szerint az árak hamarosan csökkenni fognak.
Figyelemmel kísérjük a ***LinuxBIOS*** tervezet előrehaladását is, és a jövő évre egy általa támogatott alaplapot szeretnénk majd beszerezni. Tudjuk, a türelem fontos erény, ám a néhány másodperces rendszerindítás lehetősége önmagában is vonzó.
Az idei gép hangja annyira megtetszett, hogy jövőre is csendes gépet fogunk építeni. Az adattárolás terén jövőre az ***Ed Cashin*** által a 2005 júniusi számban tárgyalt ***ATA over Ethernet*** alkalmazását is számításba fogjuk venni.
A megjelenítés továbbra is gyenge pont, de nem a hardver, hanem a gyártók jogászai miatt. Aki 3D-s megjelenítéssel foglalkozik, szinte szükségszerűen megsérti mások szabadalmait, az illesztőprogramok kódját pedig szigorú végfelhasználói szerződések védik, megtiltva a visszafejtést, és lelassítva az egész iparág fejlődését. Ha a rendszermag fejlesztése során egy széles körben használt hardverelem illesztőprogramja eltűnik, akkor vele fog menni a hardverelem is.
Grafikus kártyák gyártói, fogjatok össze, kössétek meg a hardverekre vonatkozó keresztlicencelési megállapodásokat, majd készítsetek olyan használati szerződéseket a szoftverekre és a leírásokra, amelyek alapján a fejlesztők el tudják készíteni a grafika iránt érdeklődő felhasználók által igényelt kódokat! Hosszú távon mindenki jól fog járni – például az ***NVIDIA*** kizárólag egy licencelési döntés miatt párhuzamos szoftverterjesztői rendszert tart fenn. Vajon nem tenne jót a költségvetésnek, ha ez megszűnne?
A borúlátók azt mondják, ők a realisták, és hajlamosak elfogadni a zárt illesztőprogramokat. A valóság azokban az, hogy az 1990-es évek ***UNIX***-gyártói közül egy sem támogatta a ***Linuxot***. Ma minden olyan ***UNIX***-gyártó, amely egyáltalán létezik még, a ***Linux*** mögé állt. Aki realistának hiszi magát, gondolkozzon el egy kicsit ezen.

Linux Journal 2005. 136. szám

A cikkhez tartozó források elérhetősége:
➲ ***www.linuxjournal.com/article/8330***

Justin Thiessen Linux mérnök a Penguin Computingnál. Az év linuxos erőműve tervezet vezetőjeként ő volt felelős a rendszer tervezéséért, az összeépítésért és a tesztelésért, illetve az alkatrészek kiválasztásába is bekapcsolódott. Ha nem a linuxos erőművel foglalkozik, akkor termékfejlesztési feladatokat végez, az lm_sensors tervezethez adott hozzájárulásaival pedig a Penguin hardvereinek linuxos támogatását igyekszik javítani.

Matt Fulvio szabadúszó ipari tervező és építészmérnök. A San Franciscoi Építészeti Intézetben matematikát tanít, weboldala a www.mattfulvio.com címen érhető el.

Philip Pokorny a Penguin Computing mérnökigazgatója. Az ő feladata volt a megfelelő tápegység felkutatása, valamint a vízhűtés beépítéséhez szükséges módosítások egyeztetése. Amikor éppen nem ezzel foglalkozott, akkor csak ácsorgott, és a tüsihajú főnökökre jellemző ostoba kérdésekkel bombázta a munkatársait.

Trevor Sherard a linuxos erőmű házának elkészítésében közreműködő kézműves volt, a san franciscoi öböl környékén szabadúszó szobrászként és asztalosként dolgozik. A www.woodentemple.com címen érhető el.

Don Marti a Linux Journal főszerkesztője, a cikk szövegének szerzője.

Kávéfőzés lépésről lépésre (4. rész)

Kivételkezelés és javadoc – avagy két remek érv Java- függőségünk magyarázatára

Miután az előző hónapban egy kalap alatt ismerhetett meg az Olvasó két olyan távol álló fogalmat, mint a tömbök és az interfészek, jelen írásban már nem érheti meglepetés. Ebben a cikkben egy nyelvi eszközt, a kivételkezelést, és egy, a programozásfejlesztéskor oly fontos dokumentálást segítő programot mutatok be.

A két eszköz talán egyetlen közös vonása az, hogy noha használatuk kisebb alkalmazások esetén mellőzhető, nagyobb projektek esetében nagy könnyebbséget jelentenek, és így fontosságukról ódákat lehetne zengeni.
A kivételkezelés nem kötődik szorosan az objektumközpontú filozófiához, ugyanakkor a legtöbb (ha nem az összes) objektumközpontú nyelvben megtalálható. Nincs ez másképp a Java esetében sem. Ennek a módszernek a célja a futásidőben keletkező hibák egyszerűen átlátható és kényelmes kezelése. Ezen hibák nagy részét interaktív alkalmazásokban szokás szerint a felhasználó váltja ki. Tipikus példája ennek az osztóprogram.

Divide et impera

Ebben az alkalmazásban az osztás műveletét valósítjuk meg. Egyelőre ne foglalkozzunk azzal, hogy a felhasználótól várjuk el a bemenő adatot, mindössze írjunk egy `osztas` nevű metódust.

```
/**
* Ez az osztaly az osztas muveletet valositja meg.
*/
public class Osztoprogram {

   /**
   * Az osztast vegzo metodus megjeleniti
   * a kepernyon az osztas eredmenyet.
   */
   private void osztas(double szamlalo, double
   ➥nevezo) {
      System.out.print(szamlalo + " / "
      ➥+ nevezo + " = ");
      System.out.println(szamlalo /
      ➥nevezo);
   }
   /**
   * A konstruktor harom osztast
   * vegez el.
   */
   public Osztoprogram() {
      osztas(1.0, 2.0);
      osztas(1.0, 3.0);
      osztas(1.0, 0.0);
   }

   /**
   * Az alkalmazas inditasakor letrejon egy
   * peldany az Osztoprogram osztalybol
   */
   public static void main(String[] args) {
      new Osztoprogram();
   }

}
```

Az alkalmazás belépési pontjánál egyetlen példányosítás történik. Az objektumközpontú programozásban ez megszokott gyakorlat. Ezzel a megoldással ugyanis rövid úton áthidalható az a probléma, hogy egy teljesen objektumközpontú környezetből hiányoznak a strukturált programozásban megszokott globális függvények. Ha nem hoznánk létre egyetlen objektumot sem, csak a `static` kulcsszóval minősített metódusokat használhatnánk. Természetesen megtehettük volna statikusnak az `osztas` nevű tagfüggvényt, kihagyhattuk volna a konstruktort, és törzsét átemelhettük volna a `main` függvénybe. Ekkor viszont teljes joggal feltehetnénk magunknak azt a kérdést is, hogy miért erőltetjük ezt a Java-t, ha *C*-ben szeretünk programozni.
A példányosításkor lefut az osztály konstruktora, ezért háromszor meghívásra kerül az `osztas` metódus. Ez két `double` típusú, azaz dupla lebegőpontosságú számot vár paraméterként, és ezek hányadosát íratja ki, miután a műveletet is jelezte egy sortörést mellőző `System.out.print` segítségével. Fordítás és futtatás után várakozásainkat felülmúló eredményt láthatunk a képernyőn:

```
1.0 / 2.0 = 0.5
1.0 / 3.0 = 0.3333333333333333
1.0 / 0.0 = Infinity
```

Nem meglepő a csonkított ***1/3***, ám az sokkal inkább, hogy ***1/0*** végtelen. Ezt az első ránézésre furcsa eredményt az alábbi egyenlet indokolja:

$$\lim_{n \to +0} \frac{1}{n} = +\infty$$

Azon kedves Olvasók, akik eddig sikeresen megúszták az analízis leírhatatlanul szép világának megismerését, elég ha elképzelik az ***1/n*** függvényt. Ha az ***x*** tengelyen jobbról közelítjük a nullát, a függvényérték a végtelenhez tart. Látható tehát, hogy vannak olyan matematikai eszközök, melyekkel a nullával való osztás kezelhető. Ennek ellenére az osztás műveletének ez a kivétele függvényértékként nem értelmezett, és a mindennapi gyakorlatban feltétlenül külön kell vele foglalkozni. Hogyan módosíthatnánk az `osztas` metódust, hogy megfeleljen kívánalmainknak?

```
private void osztas(double szamlalo, double nevezo) {
   System.out.print(szamlalo + " / " + nevezo + "
   ➥= ");
   if (nevezo == 0) {
      System.out.println("nem ertelmezett!");
   } else {
      System.out.println(szamlalo / nevezo);
   }
}
```

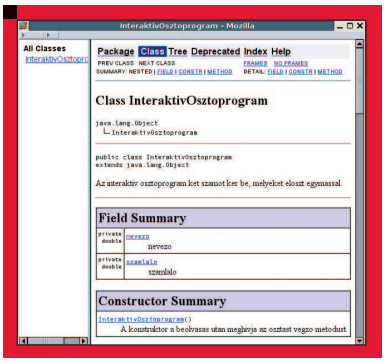
Ez a megoldás jól működik. Ám megszokhattuk, hogy ez önmagában kevés, ha a kód nem kellően beszédes. Ha egy olyan programozónak mutatnánk meg ezt a forrást, aki véletlenül nem ismeri az osztás műveletét, egy dolgot biztosan nem fog tudni eldönteni. Ebben a feltételes elágazásban melyik számít tipikus esetnek? Melyik ág az, ami normális működés mellett várhatóan lefut, és melyik az, amelyre csak egyfajta hiba előfordulásakor kerül a vezérlés? Jó volna valahogyan kifejezni ezt az ellentétet. Erre használhatjuk a `try - catch` párt.

```
private void osztas(double szamlalo, double nevezo) {
   System.out.print(szamlalo + " / " + nevezo + " =
   ➥");
   try {
      if (nevezo == 0) throw new Exception();
      System.out.println(szamlalo / nevezo);
   } catch (Exception kivetel) {
      System.out.println("nem ertelmezett!");
   }
}
```

A `try` blokk egy kritikus szakaszt hivatott jelölni. Ebben a szakaszban olyan műveletek vannak, amelyek miatt kivétel képződhet. Ez két esetben fordulhat elő. Egyrészt a programozó szándékosan dobhat kivételt a `throw` kulcsszó segítségével, amint az a példából is látszik. Másrészt vannak olyan műveletek, amelyek magukban rejtik ennek a lehetőségét. Például egész típusú (`int`) számok osztásakor, nulla értékű nevező esetén kivétel keletkezik.
A kivételek elkapása, azaz lekezelése egy `catch` ágban történik. Egy `try` blokkot több `catch` is követhet. A kivétel fellépésekor a vezérlés azonnal átkerül arra a `catch` ágra, amelyre a kivétel típusa először illeszthető. Jelen esetben egyetlen `catch` ág van, amely az `Exception`-ből, illetve az ebből örököltetett osztályokból származó példányokat kezelheti le. A `try` legelején, ha a nevező nulla, pont egy új `Exception` objektumot dobunk el. Így tehát, ha a `nevezo` változónk nulla, a `try` második sora már nem fut le, helyette az egyetlen `catch` ágban folytatódik a futás, ahol a megfelelő üzenet kerül kiíratásra.
Lássuk ennek a megoldásnak egy kicsivel pontosított változatát:

```
private void osztas(double szamlalo, double nevezo) {
   System.out.print(szamlalo + " / " + nevezo + " =
   ➥");
   try {
      if (nevezo == 0) throw new
      ➥ArithmeticException();
      System.out.print(szamlalo / nevezo);
   } catch (ArithmeticException kivetel) {
      System.out.print("nem ertelmezett!");
   } finally {
      System.out.print(System.getProperty
      ➥("line.separator"));
   }
}
```

Itt már `ArithmeticException` típusú objektumot dobunk, és kapunk el. Ez a változtatás beszédesebbé teszi a forráskódot, könnyebben kideríthető, milyen típusú hiba miatt keletkezett a kivétel. A Java kivételosztályok egész tárházát nyújtja, melyeknek közös ősosztályuk az `Exception`. Ha ezekkel nem lennénk megelégedve, készíthetnénk saját kivételosz-



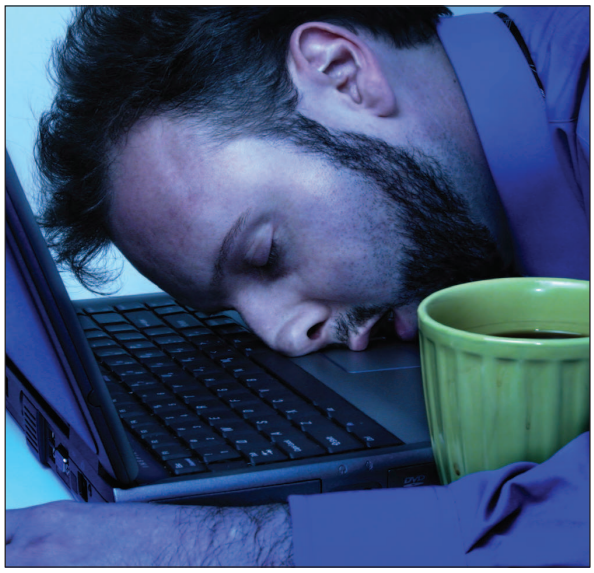
tályt is, mindössze az `Exception` osztályt kell kiterjeszteni, és máris kivételhez jutunk, amit kedvünkre dobálhatunk.
A másik változtatás a `finally` blokk bevezetése. Ebből a `try` blokkhoz hasonlóan egy lehet egy szerkezetben, ám ennek a végére kell kerülnie. Attól különleges, hogy függetlenül attól, hogy a megszokott módon, a `try` blokkban, vagy valamelyik `catch` ágban ért véget a kiértékelés, a legvégén ez kötelező érvénnyel lefut. Ebben az ágban a rendszertulajdonságok közül lekérdezzük a soremelés karakterét, és kiíratjuk. Ezzel megszabadultunk a `println` műveletektől. Így már kellően olvasmányos kódhoz jutottunk, melyet próbáljunk meg felolvasni:
Az osztás műveletéhez két számra van szükség, a számlálóra és a nevezőre. Először írassuk ki a képernyőre az elvégzendő műveletet. Majd tegyünk egy próbát a művelet elvégzésére. Ha a nevező nulla, aritmetikai hiba van, mellyel később foglalkozunk. Normális esetben el kell osztani a számlálót a nevezővel, és kiíratni. Aritmetikai hiba esetén ki kell íratni, hogy a művelet nem értelmezett. Végezetül sort kell emelni.
Sajnos egyes, a strukturált programozást sem ismerő, mégis művelő emberek azt mondják a kivételkezelésre, hogy a modern nyelvekbe így hozták vissza a `goto` parancsot. A feltétel nélküli ugrás már a strukturált programozási nyelvekben sem volt szeretett megoldás, a *Java*-ból pedig már teljesen hiányzik. Foglalt kulcsszó, tehát például `goto` nevű változót nem hozhatunk létre, de nem nyelvi elem. A kivételkezelés nem egy tiltott elemnek egy új köntösben megjelenő változata, hanem egy olyan eszköz, amivel átláthatóbbá válnak a forráskódjaink. Nem láttam még kivételkezelést, ami összezavart volna, feltétel nélküli ugrást használó spagettikódot viszont igen.
Most tegyük fel, hogy egy metódusunkban felléphet valamilyen hiba, amit nem helyben szeretnénk lekezelni. Jelen példára vetítve, az osztás műveletet kivitelező tagfüggvénytől azt várjuk el, hogy ne írjon semmit a képernyőre, hanem visszatérési értékként szolgáltassa az eredményt, és mellesleg dobjon kivételt, ha a művelet sikertelen. Lássuk a teljes forrást!

```
/**
* Ez az osztaly az osztas muveletet valositja meg.
*/
public class Osztoprogram {

   /**
   * Az osztast vegzo metodus megjeleniti
   * a kepernyon az osztas eredmenyet.
   */
   private void osztas(double szamlalo, double
   ➥nevezo) {
      System.out.print(szamlalo + " / " + nevezo +
      ➥" = ");
      try {
         System.out.print(oszt(szamlalo, nevezo));
      } catch (ArithmeticException kivetel) {
         System.out.print("nem ertelmezett!");
      } finally {
         System.out.print(System.getProperty
         ➥("line.separator"));
      }
   }

   /**
   * Ez a metodus elosztja a szamlalot a nevezovel.
   * Nulla nevezo eseten kivetelt dob, amit
   ➥a hivonak
   * kell lekezelnie.
   */
   private double oszt(double szamlalo, double
   ➥nevezo) throws ArithmeticException {
      if (nevezo == 0) throw new
      ➥ArithmeticException();
      return (szamlalo / nevezo);
   }

   /**
   * A konstruktor harom osztast vegez el.
   */
   public Osztoprogram() {
      osztas(1.0, 2.0);
      osztas(1.0, 3.0);
      osztas(1.0, 0.0);
   }

   /**
   * Az alkalmazas inditasakor letrejon egy
   * peldany az Osztoprogram osztalybol
   */
   public static void main(String[] args) {
      new Osztoprogram();
   }

}
```

A kód immár komoly esztétikai élvezetet nyújt a szakértő szemeknek. Bevezettünk egy `oszt` nevű tagfüggvényt, ami átvette az `osztas` metódus válláról a számolás terhét. Utóbbi csak kiírat, előbbi csak az eredményt számolja. Mivel az `oszt` metódusban kivétel képződhet, amit helyben nem kezelünk le, a függvény fejlécében jelezni kell, hogy a hívó felelőssége a megfelelő `try - catch` szerkezetről gondoskodni. Természetesen az is továbbháríthatja ezt a felelősséget fentebbre, ha a fejlécében szerepel a kivétel megnevezése.
Eljött az idő, hogy elkészítsük az alkalmazás felhasználóbarát változatát. Ebben már a kezelő adhatja a bemenő adatokat, vagyis nem rögzítjük a forráskódban a művelet operandusait.

```
import java.io.*;
```

```
/**
* Az interaktiv osztoprogram ket szamot
* ker be, melyeket eloszt egymassal.
*/
public class InteraktivOsztoprogram {

   /**
   * szamlalo
   */
   private double szamlalo;
   /**
   * nevezo
   */
   private double nevezo;

   /**
   * Beolvassa az adatokat a kezelotol, es
   * azonnal double tipusuva is alakitja.
   */
   private void beolvasas() {
      BufferedReader olvaso = new
BufferedReader(new InputStreamReader(System.in));
      try {
         System.out.print("szamlalo? ");
         szamlalo = Double.parseDouble
         ➥(olvaso.readLine());
         System.out.print("nevezo? ");
         nevezo = Double.parseDouble
         ➥(olvaso.readLine());
      } catch (IOException kivetel) {
         System.out.println
         ➥("Hiba tortent az adatbevitel soran!");
         System.exit(1);
      }
   }

   /**
   * Az osztast vegzo metodus megjeleniti
   * a kepernyon az osztas eredmenyet.
   * @param szamlalo
   * az osztas szamlaloja
   * @param nevezo
   * az osztas nevezoje
   */
   private void osztas(double szamlalo, double
   ➥nevezo) {
      System.out.print(szamlalo + " / " + nevezo +
      ➥" = ");
      try {
         System.out.print(oszt(szamlalo, nevezo));
      } catch (ArithmeticException kivetel) {
         System.out.print("nem ertelmezett!");
      } finally {
         System.out.print(System.getProperty
         ➥("line.separator"));
      }
   }

   /**
   * Ez a metodus elosztja a szamlalot a nevezovel.
```



```
   * Nulla nevezo eseten kivetelt dob, amit
   ➥a hivonak
   * kell lekezelnie.
   * @param szamlalo
   * az osztas szamlaloja
   * @param nevezo
   * az osztas nevezoje
   * @return az osztas eredmenye
   */
   private double oszt(double szamlalo, double
   ➥nevezo) throws ArithmeticException {
      if (nevezo == 0) throw new
      ➥ArithmeticException();
      return (szamlalo / nevezo);
   }

   /**
   * A konstruktor a beolvasas utan
   * meghivja az osztast vegzo metodust.
   */
   public InteraktivOsztoprogram() {
      beolvasas();
      osztas(szamlalo, nevezo);
   }

   /**
   * Letrehoz egy peldanyt az
   * InteraktivOsztoprogram osztalybol.
   */
   public static void main(String[] args) {
      new InteraktivOsztoprogram();
   }

}
```

Az alkalmazás igen egyszerűen működik. A `beolvasas` metódus az objektum két tagváltozójának, a `szamlalo`, és a `nevezo` tagoknak ad értéket. Ezeket felhasználva hívja meg a konstruktor az `osztas` metódust, melyet az `oszt`

segítő függvénnyel együtt a meglévő `Osztoprogram.java` forrásból emeltük át. A `beolvasas` jó példája annak, hogyan használhatjuk a kivételkezelésre építő ***Java*** könyvtári osztályokat. A `readLine` metódus `IOException` típusú kivételt dob, ha az olvasás sikertelen volt, egyébként visszaadja a beolvasott sort `String` objektumként. A `beolvasas` különlegessége még az eddig látottakhoz képest néhány új osztály használata, így többek között a `BufferedReader` és az `InputStreamReader` alkalmazása. Ezek a `java.io` csomag részei, ezért volt szükség a program legelején jelezni a csomag összes elemének behozatalát az alkalmazás névterébe. A csomagokról a későbbiekben még szó lesz, most legyen elég annyi, hogy egy osztály használatához be kell hozni azt a névtérbe. A `System` és az `Exception` osztályok esetén ezt azért kerülhettük el, mert azok a `java.lang` csomag részei, és ezért önműködően a névtér elemeit képezik.

Térjünk vissza a `Reader` osztályokhoz. Ezek segítségével olvastunk be egy-egy sort a billentyűzetről. Most ahelyett, hogy a cikk írója által adott magyarázatokból eredő kényelem miatt a kedves Olvasó hagyná, hogy elhatalmasodjon rajta a lustaság, nézzünk utána együtt ezen osztályok leírásának az API-ban. Ebből kiderül, hogy a `System.in`, ami a szabványos bemeneti csatorna, `InputStream` típusú. Ennek van ugyan `read` metódusa, ám amint az a leírásból kitűnik, bájtokat olvas és nem karaktereket. Például egy olyan terminálon, ami ***Unicode*** kódolást használ, ez nem adna használható eredményt. Ezért beburkoljuk egy `InputStreamReader`-rel, ami technikailag azt jelenti, hogy készítünk egy ilyen objektumot, és a konstruktorának átadjuk a `System.in`-t paraméterként. Ez már valóban karaktereket olvas, tehát elvégzi a szükséges átalakításokat, hogy használható bemenetet kapjunk. Ahhoz, hogy a sorvége kezelésével és a karakterek összeragasztásával se kelljen vesződnünk, ezt még beburkoljuk a `BufferedReader` osztállyal, amelynek példányai rendelkeznek az áhított `readLine` metódussal. A megoldásban az a szép, hogy a `System.in`-t egy másik adatfolyamra cserélve is tökéletesen működik, így tehát fájlból is ugyanezzel a módszerrel dolgozhatunk.

Vessünk egy pillantást a másik újdonság, a `Double` osztály leírására. Az osztálynak a `parseDouble` nevű metódusát használtuk, mindennemű példányosítás nélkül. Ezt azért tehettük meg, mert a nevezett tagfüggvény statikus. Paraméterként egy `String` típusú objektumot vár, és egy `double` típusú számot ad vissza. Meglepő ravaszsággal a szövegfüzérből egy lebegőpontos számot állít elő.

Az `osztas`, illetve `oszt` műveletek megvalósításai nem változtak, mindössze a megjegyzések egészültek ki egy-két érdekes elemmel. Ennek oka abban keresendő, hogy az Olvasóban az ***API*** tüzetes tanulmányozása után biztosan felmerült a kérdés, hogy miként tudna hasonlóan jól áttekinthető, egységes leírást mellékelni ***Java*** alkalmazásaihoz. A válasz egyetlen parancssorral megadható:

```
$ javadoc -d docs -private
➥InteraktivOsztoprogram.java
```

A parancsot abban a könyvtárban kell kiadni, ahol a forráskód található. A `-d` kapcsoló azt a célkönyvtárat határozza meg, amelyben a ***HTML*** alapú dokumentációt szeretnénk látni. A `-private` hatására a személyes láthatóságú adattagok is bekerülnek a leírásba, ami azért fontos, mert nem ez az alapértelmezés. A kapcsolók után azon megfelelő formátumú megjegyzésekkel tűzdelt forrásokat kell felsorolni, amelyekből ki szeretnénk nyerni az értékes információt. Jelen esetben ez egyetlen állomány.

Rövid várakozást követően elkészül a `docs` könyvtár, melyből az `index.html` lapot megnyitva kedvenc böngészőnkből a képen látható tartalom tárul elénk. Hasonlítsuk össze a leírást a forráskód megjegyzéseivel, és máris látni fogjuk, miből lesz a cserebogár. A dokumentációban is megjelenő megjegyzések kötelezően `/**` és `*/` között szerepelnek, és mindig a hivatkozott elem előtt. Metódusok esetén érdemes kihasználni a címkék nyújtotta lehetőségeket. A címkéket `@` jel vezeti be, és így kényelmesen megadható egy tagfüggvény paramétereinek és visszatérési értékének rövid magyarázata.

Ennyit terveztem erre a hónapra. Ha összehasonlítjuk az alcímben is szereplő két témának a súlyát a cikkben, meglepődve tapasztalhatjuk, mennyire részrehajlóan sok szó esett az elsőről. Ennek oka az, hogy a ***javadoc*** egy az egyszerű, és mégis nagyszerű eszközök közül. Érdemes egy kis időt eltölteni a kapcsolók listájának böngészésével, ám kezdetnek tökéletesen elég, ha gondosan karban tartjuk megjegyzéseinket, mert így pillanatok alatt kiváló dokumentációt varázsolhatunk.

Sok örömet kívánok a kísérletezéshez, a sorozattal kapcsolatban pedig várom az észrevételeket, javaslatokat.

Fülöp Balázs (bigwig42@gmail.com)
21 éves, imádja a Túró Rudit, a Debian Linuxot és a teheneket. Kedvenc írója Slawomir Mrożek. Leginkább a számítógépes hálózatok biztonsága érdekli. A BME VIK műszaki informatikus szak hallgatója.

Virtuális gépek kialakítása XEN segítségével

Előbb-utóbb minden rendszergazda szembetalálkozik azzal a helyzettel, amikor újabb kiszolgálók beüzemelésével bízzák meg, csak éppen elfelejtenek hozzá új gépet adni. Paradoxonnak tűnik, ugye? Vagy mégsem? Ilyenkor jöhet jól egy olyan szoftver, amelynek segítségével megoszthatjuk a már rendelkezésünkre álló kiszolgálók erőforrásait.

Xen virtual machine monitor (VMM)

Az ***x86***-os architektúrára jelenleg két elterjedt ***virtualizálási technológia*** létezik. Az egyik a vendég operációs rendszerek módosítás nélküli futtathatóságát helyezi előtérbe (***VMware***), míg a másik a nagyobb teljesítményt (***Xen***, ***Denali***). Az első megközelítés kétségtelen előnye, hogy szinte minden x86 architektúrára készült rendszert képes futtatni, de ennek az ára a relatív kisebb teljesítmény. A második módszer a vendég operációs rendszerből majdnem natív teljesítményt hoz ki, viszont igényli azok módosítását kernel szinten. Ezt a virtualizálási technológiát ***paravirtualizációnak*** hívják. Ezt a megközelítést alkalmazza a ***Xen*** is.

A ***Xen VMM*** a ***XenoServers*** projekt részeként született meg a ***University of Cambridge Computer Laboratory*** berkein belül. A ***Xen***, definíciója szerint: paravirtualizáló virtuális gép monitor x86 architektúrára. A paravirtualizáció a vendég operációs rendszerek számára az alattuk lévő hardver eléréséhez egy szoftver interfészt biztosít. Ez az interfész működésében nagyon hasonlít az eredeti hardverhez, de különbözőségek is akadnak szép számmal. Ezeket az eltéréseket úgy oldotta fel a fejlesztő csapat, hogy a ***Xen***-ből egy külön ***arch***-ot készítettek, mintha a ***Linux*** kernelt egy új platformra kellett volna átültetni. További előnye ennek a rendszernek, hogy a felhasználói programok változtatás nélkül futtathatók. A rendszer struktúráját lásd az *1. ábrán*.

A ***GNU/Linux*** változat mellett jelenleg fejlesztés alatt állnak a ***NetBSD***, ***FreeBSD***, ***Plan 9*** portok is. Mindezek mellett a fejlesztés korai szakaszában elkészült a ***Windows XP*** verzió is, amelyet – sajnos – licencelési problémák miatt nem lehet a nagy közönség számára elérhetővé tenni. Ebben az ügyben előrelépés akkor várható, amikor az ***Intel*** és az ***AMD*** által bejelentett új technológiák végre megjelennek az új processzoraikban (***Intel® Virtualization Technology*** – ➲ ***http://www.intel.com/technology/computing/vptech/*** ; ***AMD Pacifica*** – ➲ ***http://www.amd.com/us-en/Corporate/VirtualPressRoom/0,,51_104_543~98372,00.html***).

A ***Xen x86*** architektúrájú, ***„P6"*** vagy annál újabb processzorokat igényel (***Intel Pentium Pro***-tól ***Xeon***-ig, illetve ***AMD Athlon***, ***Duron***). Támogatja az ***SMP***-t és hajlandó működni ***Hyper-Threading*** képes processzorokon is. A 64 bites verzió fejlesztés alatt áll, habár már most is hajlandó elindulni ilyen gépeken 32 bites módban. Tervezik továbbá a ***PPC*** és ***ARM*** architektúrák támogatását és a memória használat jelenlegi korlátainak bővítését. A ***Xen*** telepíthető notebookra is, de készüljünk fel az akkumulátorok gyors merülésére, mivel sem az ***APM*** sem az ***ACPI*** nem támogatott.

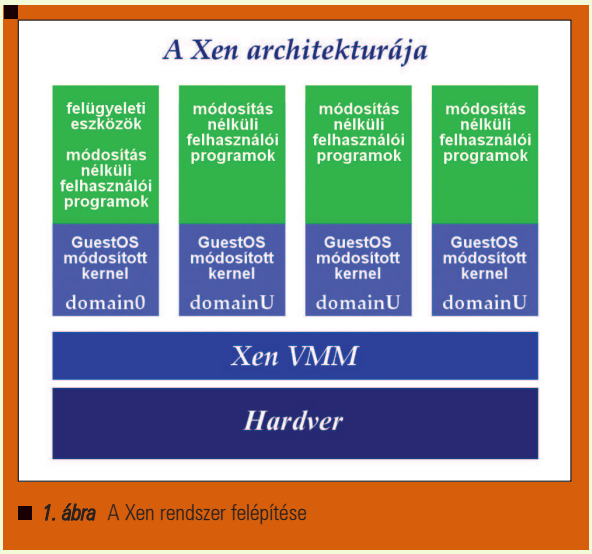
A teljesítményről bővebben a ➲ ***http://www.cl.cam.ac.uk/Research/SRG/netos/xen/performance.html*** weboldalon találunk információt, összehasonlító táblázatokat:

Telepítés

A telepítés menetét ***Debian Sarge***-ra fogom leírni, de más disztribúciókon is hasonló lépéseken kell majd végigmennünk. Lássuk milyen csomagok érhetőek el a projekt web oldalán: először is van a stabil verzió, amely a cikk írásának pillanatában a 2.0.6-os verziónál tart, találhatók továbbá ***testing*** és ***unstable*** verziók is. Természetesen elérhető a program forrása, és a 64 bites verziót is le lehet tölteni. Aki a telepítés nehézségei nélkül szeretné megtekinteni a programot az se keseredjen el, hiszen a ***Xen 2.0.6 Demo CD*** segítségével megteheti. Csak le kell tölteni a ***xendemo-2.0.6.iso***-t, kiírni egy ***CD***-re és egy bootolás után máris láthatjuk a saját gépünkön békésen egymás mellett élni a ***Linux 2.4*** vagy ***2.6*** kerneles verzióit, a ***NetBSD***-t és a ***FreeBSD***-t.

Ha megtetszett amit láttunk és kedvünk lenne telepíteni, akkor töltsük le a ➲ ***http://www.cl.cam.ac.uk/Research/SRG/netos/xen/downloads.html*** oldalról a stabil verziót.

A telepítés történhet előre fordított binárisból vagy forrásból. Szerencsére egyik sem túl bonyolult. Mindenek előtt olvassuk el a csatolt dokumentációkat. Tudom, hogy mindig



1. ábra A Xen rendszer felépítése

mindenki ezt írja, de ez talán nem véletlen. A felhasználói kézikönyv elég részletes, minden szükséges információt tartalmaz közérthető módon tálalva és szerencsére a projekt web oldalán is elérhető (➲ *http://www.cl.cam.ac.uk/Research/SRG/netos/xen/readmes/user/user.html*).
Amennyiben úgy gondoljuk, hogy minden érthető kezdjük a telepítést a binárissal. A letöltött állományt tömörítsük ki, majd az telepítőszkripttel telepítsük:

```
# tar -xvzf xen-2.0.6-install-x86_32.tgz
# cd xen-2.0
# ./install.sh
```

A folyamat végeztével a bináris állományt feltelepítettük. Ha a nagyobb kontroll kedvéért a forrást választanánk a következő lépések várnak ránk:

```
# tar -xvzf xen-2.0.6-src.tgz
# cd xen-2.0
# make world
# make install
```

A telepítő le fogja tölteni a *kernel.org*-ról a neki szükséges kernel forrását, ráteszi a saját foltjait, majd elvégzi a fordítást és a helyükre másolja az állományokat. Előfordulhat, hogy a telepítés megáll valamilyen fájl hiányára panaszkodva. *Debian* rendszeren a következő csomagok megléte szükséges:

```
# apt-get install make gcc libc6-dev bzip2
module-init-tools latex latex2html transfig tgif
bridge-utils zlib1g zlib1-dev python python-dev
python-twisted iproute libcurl3 libcurl3-dev
```

A későbbiek során szükségünk lehet a kernel újrafordítására. Ehhez lépjünk be a *Xen* forrásában található *linux-2.6.XX-xen0* könyvtárba, majd konfiguráljuk a *Xen arch*-ra a kernelt:

```
# cd linux-2.6.XX-xen0
# make ARCH=xen menuconfig
# cd ..
# make dist
# make install
```

Mind a bináris mind a forrásból telepített *Xen* állományokat helyez el a következő könyvtárakba:
/boot – a Xen; *domain0* és *domainU* kernelek;
/etc/xen – konfigurációs állományok
/etc/init.d – indító szkriptek
/lib/modules – kernel modulok
/usr – felügyeleti eszközök, dokumentációk, program könyvtárak, *Python* állományok

Amennyiben a használat során a rendszer megmagyarázhatatlannak tűnő lassulását észlelnénk, tiltsuk le a *TLS (Thread Local Storage)* program könyvtárat, mert jelenlegi verziójában nem kompatibilis a *Xen*-el.

```
# mv /lib/tls /lib/tls.disable
```

Amennyiben mégis hiányát éreznénk bármikor helyre tudjuk állítani az eredeti állapotot a könyvtár vissza nevezésével. Már csak egyetlen lépés maradt hátra, módosítanunk kell a *GRUB* konfigurációs állományát (a *Xen* a *GRUB* rendszerbetöltő meglétét igényli), hogy a frissen telepített *XenLinux* kernelről induljon a rendszerünk. Adjuk hozzá a `/boot/grub/menu.lst` állományhoz a következő sorokat:

```
title Xen 2.0 / XenLinux 2.6
  kernel /xen.gz dom0_mem=131072
  module /vmlinuz-2.6-xen0 root=/dev/hdXY ro
  ➥console=tty0
```

ahol `/dev/hdXY` helyére írjuk be a rendszerünk root partíciójának nevét. Ha mindezzel készen vagyunk indítsuk újra a gépet és győződjünk meg róla, hogy a megfelelő bejegyzést válasszuk a *GRUB* indító menüjéből.

Felügyeleti eszközök

A bootfolyamat során előbb a *Xen* néhány soros üzenetei láthatóak, aztán elindul a *XenLinux* kernel a *domain0* virtuális gépen. A *Xen* terminológiában a vendég operációs rendszereket *domain*-eknek hívjuk. A különböző domain-ek között kiemelt szerepet kap a domain0, amely felügyeleti jogokkal rendelkezik. Ez a szerkezeti felépítés látható az *1. ábrán*.
A *domain0*-án keresztül szabályozhatjuk a *Xen* működését a rendelkezésünkre álló felügyeleti eszközökkel. A felügyeleti tevékenységek nagy részét az *xm* nevű eszközzel fogjuk végezni. Az `xm` parancs megadási formája a következő:

```
# xm utasítás [kapcsolók] [argumentumok] [változók]
```

Lássuk az `xm` legfontosabb utasításai:
`xm list` – kilistázza a futó domain-eket;
`xm consoles` – információt nyújt a domain konzolokról;
`xm console` – konzolt nyit a paraméterként megadott azonosítóval (ID) rendelkező domain-hez;
`xm save` – elmenthetünk fájlba egy futó domain-t a domain0 fájlrendszerébe (a Debian Sarge nem tartalmazza a `libcurl.so.2` fájlt, amire a `save` utasításnak szüksége van, ezért készítsünk ilyen néven egy symlinket a `/usr/lib/libcurl.so.3.0.0` fájlról);
`xm restore` – az `xm save`-vel elmentett domain-t tudjuk ezzel a paranccsal elindítani;
`xm migrate` – ezzel a paranccsal lehetőségünk van egy xen-t futtató szerverről átmásolni egy domain-t egy másik

xen-t futtató gépre, akár úgy, hogy az adott domain-en futó szolgáltatások felhasználói ebből gyakorlatilag ne vegyenek észre semmit;
`xm create` – létrehoz egy új domain-t;
`xm shutdown` – leállít egy domain-t.

Az utasítások pontos paraméterezését megtalálhatjuk a dokumentációban vagy az `xm help utasítás` parancs kiadásával. Miután birtokában vagyunk az alapvető kezelési ismereteknek nézzük mit kell tudnunk a vendég operációs rendszerekről.

Megérkezik az első vendég

Példaként telepíteni fogunk egy ***Debian Sarge*** rendszert `debootstrap` segítségével, ezért telepítsük is fel:

```
# apt-get install debootstrap
```

Ha még nem tettük volna meg, indítsuk el a ***Xen*** démont:

```
# /etc/init.d/xend start
```

Az `xm list` paranccsal ellenőrizzük a futó domain-eket.

```
# xm list
Name      d  Mem(MB)  CPU  State  Time(s)  Console
Domain-0 0       123    0  r----      9.0
```

Látható, hogy a domain0 már fut, és csak annyi területet foglal a rendszermemóriából, amennyit részére a ***GRUB*** konfigurációs állományában beállítottunk. Vendég operációs rendszerünk viszont jól láthatóan még nincs. Hozzunk akkor létre egyet. Amennyiben van üres terület a merevlemezünkön egyszerű a helyzetünk. Alakítsunk ki két partíciót, egyet a root és egyet a swap számára, majd formázzuk meg őket. A cikk további részén helyettesítsük a példákba a létrehozott két partíció útvonalát ott, ahol az szükséges. Mit tegyünk viszont akkor, ha nincs elég szabad területünk? Természetesen használhatjuk valamelyik linuxos programot, hogy átméretezzük a meglévő partícióinkat, vagy kihasználhatjuk a xen azon lehetőségét, hogy a fájlokban létrehozott blokk egységeket is képes kezelni. Készítsünk egy-egy fájlt a root és a swap partíciók részére:

```
# dd if=/dev/zero of=/opt/domain1/root bs=1024k
➥count=2048
# dd if=/dev/zero of=/opt/domain1/swap bs=1024k
➥count=256
```

Az első paranccsal létrehoztunk egy 2048 megabyte-os fájlt „root” néven a root partíció részére az */opt/domain1* könyvtárban, míg a második paranccsal egy 256 megabyte-os fájlt ”swap” néven a swap partíciónak. Készítsük el a fájlrendszert ezekben az állományokban majd csatoljuk fel a root partíciót a loop eszközön keresztül:

```
# mkfs.reiserfs /opt/domain1/root
# mkswap /opt/domain1/swap
# mount -o loop /opt/domain1/root /mnt
```

Kezdjük el a rendszer telepítését a `debootstrap` segítségével:

```
# debootstrap --arch i386 sarge /mnt
➥http://ftp.hu.debian.org/debian
```

Amikor befejeződött a telepítés lépjünk be az új rendszerünkbe:

```
# chroot /mnt /bin/bash
```

Módosítsuk az új rendszer igényeinek megfelelően a következő konfigurációs állományokat:

```
# /etc/resolv.conf
# /etc/hostname
# /etc/hosts
# /etc/network/interfaces
# /etc/networks
# /etc/apt/sources.list
```

Az első három fájl kitöltésével nem lehet problémánk, de időzzünk el egy kicsit a hálózati beállításoknál. Mielőtt hozzá kezdenénk el kell döntenünk néhány fontos kérdést. Először is: szükségünk van-e egyáltalán hálózatra? Első hallásra talán butaságnak tűnhet a kérdés felvetése, de nem az. Habár a cikk címe szerint szervereket fogunk majd üzemeltetni a virtuális gépeken, de kifejezetten ajánlják például kernel fejlesztéshez is. Ebben az esetben valószínűleg tényleg nem lesz szükségünk hálózatra, hiszen a virtuális gépet elérhetjük az `xm` parancs `console` funkcióján keresztül. Ha maradunk a cikk eredeti irányvonala mentén és szervert készülünk telepíteni virtuális gépünkre, akkor viszont minden bizonnyal szükségünk lesz hálózatra. Ezzel el is érkeztünk a második megválaszolandó kérdéshez: bridge-elt vagy bridge-elt/routeolt hálózatot fogunk használni? A xen mindkét esetben virtuális hálózati interfészt hoz létre a virtuális gépekben, a különbség ezek működésében keresendő. Az első esetben az adott virtuális gép virtuális hálókártyáján ugyanazok a csomagok fognak megjelenni, mint a hálózathoz fizikailag kapcsolt kártyán. Minden virtuális kártya saját külső IP és MAC címmel fog rendelkezni, mintha az adott hálózathoz fizikailag is több gép lenne csatlakoztatva. A második esetet képzeljük el úgy, mintha az összes virtuális gépet egy router-en keresztül csatlakoztatnánk a hálózathoz. Válasszuk tehát az első lehetőséget akkor, ha minden virtuális gépünk saját külső IP címmel fog rendelkezni, a másodikat pedig akkor, ha a külvilág számára az összes gép egy IP címen lesz elérhető.
A bridge-elés a xen telepítésénél az alapbeállítás. Tételezzük fel, hogy a ***domain0***-ás gépünk külső IP címe 1.2.3.4, az 1.2.3.0/24-es hálózathoz tartozik és átjárója 1.2.3.254. Ebben az esetben a ***domain1*** hálózati kártyája részére osszunk ki egy szabad IP címet a 1.2.3.0/24-es tartományból és ezzel készen is vagyunk. Legyen az `/etc/network/interfaces` fájl tartalma például ez:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
  address 1.2.3.100
  network 1.2.3.0
  netmask 255.255.255.0
  broadcast 1.2.3.255
  gateway 1.2.3.254
```

LVM létrehozása RAID tömbön

```
# pvcreate /dev/md1                          # inicializáljuk a partíciót
# vgcreate storage /dev/md1                  # létrehozzuk a "storage" nevu volume group-ot a /dev/md1 raid tömbön
# lvcreate -L4096M -n dom1root storage       # létrehozunk egy 4GB-os partíciót  "dom1root" néven
# mkfs.reiserfs /dev/storage/dom1root        # leformázzuk az új partíciót
```

Partíció méretének csökkentése

```
# resize_reiserfs -s -1G /dev/storage/dom1root     # csökkentjük a fájlrendszer méretét 1GB-al
# lvreduce -L -1G storage/dom1root                 # majd csökkentjük a partíció méretét is 1GB-al
```

Partíció méretének növelése

```
# resize_reiserfs -s +1G /dev/storage/dom1root     # növeljük a fájlrendszer méretét 1GB-al
# lvextend -L +1G storage/dom1root                 # majd növeljükjük a partíció méretét is 1GB-al
```


Kicsit bonyolódik a helyzet, amennyiben a második lehetőséget választjuk. Jelöljünk ki egy hálózati tartományt a virtuális gépek részére, legyen mondjuk a 192.168.1.0/24-es. Végezzük el a szükséges módosításokat a ***domain0*** konfigurációjában. Adjunk hozzá az `/etc/network/interfaces` fájlhoz még egy interfészt:

```
auto xen-intbr
iface xen-intbr inet static
   pre-up brctl addbr xen-intbr
   post-down brctl delbr xen-intbr
   address 192.168.1.254
   network 192.168.1.0
   netmask 255.255.255.0
   broadcast 192.168.1.255
   bridge_fd 0
   bridge_hello 0
   bridge_stp off
```

Módosítsuk az `/etc/xen/xend-config.sxp` állományt a következő módon:

```
## Use the following if VIF traffic is routed.
# The script used to start/stop networking for xend.
(network-script network-route)
# The default script used to control virtual
➥interfaces.
#(vif-script vif-route)

## Use the following if VIF traffic is bridged.
# The script used to start/stop networking for xend.
#(network-script network)
# The default bridge that virtual interfaces should
➥be connected to.
(vif-bridge xen-intbr)
# The default script used to control virtual
➥interfaces.
(vif-script vif-bridge)
```

A ***domain1*** hálózati beállításait végezzük el a következőképpen:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
   address 192.168.1.1
   network 192.168.1.0
   netmask 255.255.255.0
   broadcast 192.168.1.255
   gateway 192.168.1.254
```

Ahhoz, hogy működjön a NAT-olás már csak egy teendőnk maradt, engedélyeznünk kell a csomagok továbbítását a ***domain0***-án.

```
iptables -A POSTROUTING -o eth0 -s 192.168.1.0/24
➥-j SNAT --to-source 1.2.3.4
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Ha a későbbiek során elérhetővé szeretnénk tenni valamely szolgáltatást a virtuális gépen, a tűzfalat kell módosítanunk. Példa: a ***domain1*** ssh szerverének elérése a külső IP címen és a 20022-es porton:

```
iptables -A PREROUTING -p tcp -m tcp -d 1.2.3.4 -
➥-dport 20022 -j DNAT --to-destination 192.168.1.1
```

A kis hálózati kitérő után folytassuk az első virtuális gép konfigurálását. Hozzuk létre az */etc/fstab* állományt a következő tartalommal:

```
/dev/sda1  /       reiserfs         errors=remount-ro  0  1
/dev/sda2  none    swap  sw         0      0
proc       /proc   proc  defaults   0      0
```

Ne lepődjünk meg az `sda1` és `sda2` partíciókon, a virtuális gépek így fogják látni a számukra kiajánlott partíciókat. Lépjünk ki a ***chroot***-ból, majd csatoljuk le az */opt/domain1/root* eszközt.

```
# exit
# umount /mnt
```

Elkészült az első vendég operációs rendszerünk. Hozzunk létre számára az `/etc/xen/domain1.conf` konfigurációs állományt. Használjuk bátran a minta állományokat, legyen a tartalma például valami hasonló:

```
name="Domain-1"
memory=128
kernel="/boot/vmlinuz-2.6-xenU"
nics=1
disk = ['file:/opt/domain1/root,sda1,w',
```

```
➥'file:/opt/domain1/swap,sda2,w' ]
root="/dev/sda1 ro"
vif = ['mac=00:06:AA:B5:C1:76']
```

Figyeljük meg a `disk` sort, itt állítjuk be azt, hogy a domain0-ban létrehozott eszközök a virtuális gép számára milyen partícióként látszódjanak. Amennyiben különálló partíciókat, vagy a későbbiekben ajánlott ***LVM***-ben létrehozott partíciókat használunk, a sort módosítsuk ilyen formán:

```
disk=['phy:/dev/storage/dom1root,sda1,w','phy:/
➥dev/storage/dom1swap,sda2,w']
```

Említést érdemelnek még a `memory` és a `vif` kezdetű sorok. Az előző az új virtuális gép által használható memória mennyiségét határozza meg, míg a második a virtuális gép virtuális hálókártyájának állít be ***MAC*** címet. Ez kifejezetten előnyös, az egyébként használatos automatikusan generált címekkel szemben. Elkerülhetőek így például a változó címekből adódó ***ARP*** feloldási problémák.
Elérkeztünk a nagy pillanathoz, indítsuk el első virtuális gépünket.

```
# xm create -f /etc/xen/domain1.conf
```

Amennyiben az előzőekben elfelejtettük volna lecsatolni a root partíciót, most hibajelzést kapnánk. A ***Xen*** figyel arra, hogy ugyanaz az eszköz ne legyen két rendszerben is írásra felcsatolva, mert az súlyos adatvesztést okozhat. Ilyen esetben figyelmeztetést kapunk és nem indítja el a domain-t. A log fájlokban egyébként bőségesen kapunk visszajelzést az esetlegesen felmerült hibákról. Ha mindent jól csináltunk nem kapunk hibaüzenetet és elindul virtuális rendszerünk.

```
Using config file "/etc/xen/domain1.conf".
Started domain Domain-1, console on port 9601
```

Ellenőrizzük újra a futó domain-ek listáját:

```
# xm list
Name      Id  Mem(MB)  CPU  State  Time(s)  Console
Domain-0  0      123    0   r----     15.9
Domain-1  1      127    1   r----    127.7    9601
```

Remekül látható, hogy fut az első vendég "Domain-1" néven 1-es ID-val. Csatlakozzunk az új rendszerhez:

```
# xm console 1
```

Kezdjük el friss ***Debianunk*** beállítását, erre használjuk a rendszer nyújtotta `base-config` segédprogramot. Ezután adjunk jelszót a root felhasználónak.

```
# /usr/sbin/base-config
# passwd
```

Ha még nem tettük volna meg, állítsuk be a hálózatot. Módosítsuk az `/etc/network/interfaces` fájlt a hálózatunk igényeinek megfelelően a korábbiakban leírtak alapján.

Frissítsük a rendszert az internetről és telepítsük fel az ***ssh*** szervert a könnyebb felügyelhetőség érdekében:

```
# apt-get update
# apt-get upgrade
# apt-get install ssh
```

Amikor készen vagyunk az alapvető beállításokkal lépjünk ki a virtuális gépből egyszerre megnyomva a ***Ctrl*** és ***]*** gombokat. Már csak egyetlen dolgunk maradt a végére, biztosítsuk, hogy a ***Xen*** démon elinduljon minden újraindulás alkalmával és elindítsa a vendég gépeket is:

```
ln -s /etc/init.d/xend /etc/rc2.d/S19xend
ln -s /etc/init.d/xend /etc/rc0.d/K90xend
ln -s /etc/init.d/xend /etc/rc6.d/K90xend
ln -s /etc/init.d/xendomains
/etc/rc4.d/S99xendomains
ln -s /etc/init.d/xendomains
/etc/rc0.d/K10xendomains
ln -s /etc/init.d/xendomains
/etc/rc6.d/K10xendomains
ln -s /etc/xen/domain1.conf
/etc/xen/auto/domain1.conf
```

Konklúzió

Térjünk vissza egy gondolat erejéig a cikk elején idézett definícióhoz. Ahogy látható a ***Xen*** valódi paravirtualizálást végez, nagy számú párhuzamosan futtatható virtuális szervert bírhatunk működésre ugyanazon a hardver eszközön és azok működését teljes körűen felügyelhetjük (monitorozhatjuk) a felügyeleti eszközök segítségével. Amennyiben szervereket szeretnénk üzemelni a létrehozott tartományokban, akkor valószínűleg fontos az adatbiztonság és a rugalmasság. Ha lehetőségünk van rá, akkor használjunk több merevlemezt, hozzunk létre ***RAID 5*** tömböt a vendég operációs rendszereknek, majd azon ***LVM*** segítségével alakítsuk ki a szükséges partíciókat (a szükséges lépéseket lásd az keretes listában). Ezzel a módszerrel egy könnyen konfigurálható és biztonságos fájlrendszert tudunk a ***Xen*** alá rakni, ami megfelel a fent említett igényeknek.
Saját tapasztalataim alapján azt mondhatom, hogy kiváló eszközt kapnak a rendszergazdák a kezükbe akár tesztelésre, akár éles üzemben való használatra. A rendszerek elmenthetőségének és visszaállításának, valamint a migrálásnak a lehetősége kiemelt rugalmasságot biztosít. Értékét tovább emeli a stabil, megbízható működés, és a – számunkra oly kedves – nyílt forrás, hiszen a szoftver teljes forrása ***GNU GPL*** védelme alatt érhető el.

Hábit József (Habit@osi.hu)
Ha mások kérdeznék, imádok sportolni, túrázni és főzni. Gyakorlatilag viszont zseniálisan kezelem a távirányítót és felnőtt fejjel sem vetem meg a számítógépes játékokat. Szeretem a focit és a sört!

Gép a gépben – VMware

Bizonyára az olvasóval is előfordult már, hogy szeretett volna kipróbálni egy új Linux disztribúciót vagy bármilyen más operációs rendszert egyszerre használni jelenlegivel, de épp nem volt kéznél egy fölösleges számítógép, amire nyugodt szívvel lehetett volna telepíteni.

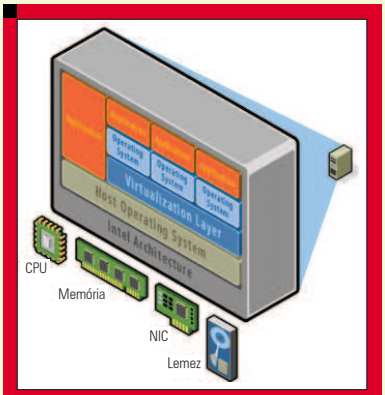
Ilyenkor általában két lehetőségünk van: szerelhetünk egy másik merevlemezt számítógépünkbe, vagy átparticionálhatjuk a már benne lévőt. Így telepíthetünk több operációs rendszert is egyetlen számítógépre. Ezzel a módszerrel megnézhetünk például olyan weboldalakat amelyek csak ***Internet Explorerrel*** hajlandóak működni, bár egy idő után zavaró lehet az állandó váltogatás a rendszerek között.
Ha a fentiek nem felelnek meg esetleg lehet próbálkozni *PC* emulátor telepítésével is. Ha az olvasónak a harmadik megoldás a legszimpatikusabb, ajánlom ne hagyja abba az olvasását, ki fog derülni mi is az a *PC* emulátor, pontosan mire jó, és egy konkrét példán keresztül be mutatom a használatát, természetesen egy cikk adta lehetőségek között.

PC emulátorok

Tapasztaltabb felhasználók mondhatnák hogy a fenti célra tökéletesen megfelel egy *Windows* emulátor – ilyen a *WINE* –, azzal lehet futtatni *Internet Explorert*, sőt még sokkal bonyolultabb programokat, játékokat, szövegszerkesztőket is. Igen ez így van, ezért a fenti példa egy kicsit sántít is, de talán gondoljunk az egyidejűség kérdésére. Ennek akkor van szerepe ha egy hálózati szoftvert, például fájlszervert szeretnénk tesztelni.
Itt jön a képbe a ***host-only network***, amelyről még lesz szó.
Konkrét példaprogramunk is lesz méghozzá a *VMware*. A szoftvert a *VMware Inc.* fejleszti 1998-óta, az első változat 1999-ben látott napvilágot. Később – az eredeti célkitűzésnek megfelelően – tovább bővült a termékek száma, mára egész virtuális infrastruktúrát sikerült kifejleszteni.
Mi az asztali gépekre szánt *VMware Workstation* (továbbiakban csak *VMware*) nem rég megjelent 5.0.0-ás változatával fogunk behatóbban megismerkedni, akit érdekelnek részletesen a *VMware Inc.* termékei látogasson el a cég honlapjára (lásd a Kapcsolódó címek részt), érdemes megtenni.
A kategória másik vezető terméke a *Microsoft Virtual PC*, ez egyenlőre csak *Windowson* és *Mac OS X*-en fut. Mindkét program kereskedelmi, vagyis zárt forrású termék, aki azonban ki szeretné kipróbálni bármelyiket, kérhet ***bemutató licencet (evaluation serial number)*** a gyártótól.

VMware virtualizációs technológia

Itt az idő hogy beleássuk magunkat a mélyebb részletekbe, technológia működését az *1. ábra* szemlélteti.
Ezen legalul látható a fizikai hardverünk (*Intel Architecture*), a *VMware* egyenlőre csak az *IBM PC* – másik nevén *x86* – típusú számítógépeket támogatja. Erre telepítjük ***gazdarendszerünket (Host Operating System)***, ez *Windows* vagy *Linux lehet VMware* esetén.
A következő réteg a ***virtualizációs réteg (Virtualization Layer)***, itt történik az emuláció, ebben a rétegben fut a *VMware* egyszerű felhasználói programként, ez végzi az emulációját bizonyos hardvereknek. Ezen a ponton kell szót ejtenünk ezekről a „bizonyos hardverekről”, ugyanis a *VMware* nem



1. ábra A VMware virtualizációs technológia felépítése

minden hardvert emulál, így tesz például a processzorral (64-bites processzorokon 32-bites módban futtatja az alkalmazásokat és az operációs rendszert), a monitorral és a memória egy részével is, néhány hardverről dönthetünk – többek közt ilyen a merevlemez és a ***CD-ROM*** –, némelyeket feltétlen emulál, például videóvezérlőt és hangkártyát. Ezekre a lehetőségekre még visszatérek a gyakorlati résznél.
A következő rétegek (***Operating System, Aplications***) már az emulált hardveren futnak.
A *VMware*-re telepített operációs rendszer teljes mértékben „azt hiszi” hogy egy valóba létező hardver az, amin fut, tehát úgy érzi, hogy a processzor csak az övé. A virtuális hardverre telepített operációs rendszert, ***vendégrendszernek (Guest Operating System)*** nevezzük. Egy időben több virtuális gép is létezhet

és párhuzamosan futhat is, ezek között lehet hálózati kapcsolat, egymást valóban létező számítógépnek hiszik.

VMware telepítése és beállítása

Most hogy már képet kaptunk a *VMware*-ről, itt az idő, telepítsük számítógépünkre. Rendszerkövetelménye nem túl magas, a felhasználói kézikönyv 500 MHz vagy annál gyorsabb x86-os processzort javasol 256 MB memóriával, nem számítva a vendégrendszerek memóriaigényét. A telepítéshez töltsük le a *VMware Workstation* telepítőjét a *VMware Inc.* honlapjáról, ezt csak regisztráció és egy kérdőív kitöltése után tehetjük meg. Mielőtt használatba vehetnénk a *VMware*-t két dolgot kell megtennünk, installálni majd konfigurálni. Előbbi a ténylegesen fájlok telepítéséből utóbbi a már föltelepített szoftverrendszer beállításából áll. A telepítőből létezik *rpm* és *tar.gz* csomag, röviden bemutatom mindkettő telepítését, ezután pedig a konfigurálást.
Ha *rpm* alapú rendszerünk van akkor vagyunk könnyebb helyzetben, ekkor az

```
rpm -Uhv Vmware-workstation-
➥5.0.0-13124.i386.rpm
```

parancs kiadásával telepíthetjük, ha a csomag néhány függősége netán nincs föltelepítve installáljuk azokat, majd próbálkozunk ismét. A konfigurációs szkriptet a

```
vmware-config.pl
```

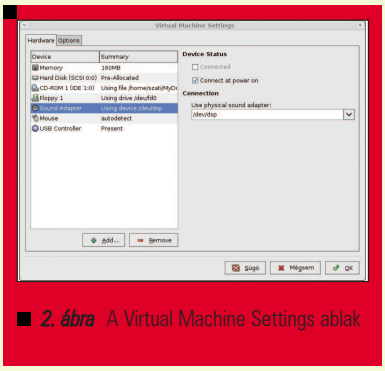
paranccsal indíthatjuk el.
Aki a *tgz* csomagot választja, vagy nem *rpm* alapú rendszerre telepít, annak kicsit nehezebb dolga lesz ugyanis saját magának kell ellenőrizni a telepítéshez szükséges csomagok meglétét, és néhány információt a telepítő szkript számára megadni. Legelőször is *tar* és a *gzip* programok szükségesek a tgz archívum kicsomagolásához (`tar -vzxf Vmware-Worksation-5.0.0-13124.tar.gz`). Ha ez megvan akkor lépjünk be a *vmware-distrib* könyvtárba, váltsunk át rendszergazda felhasználóra (`su root`), majd futtassuk a *vmware-install.pl* szkriptet (`./vmware-install.pl`). Néhány kérdést tesz föl a telepítő, ezek a szoftver különböző részeinek telepítési helyére vonatkoznak, nem túl sok értelme van ezeket megváltoztatni, nyugodtan maradhat az alapbeállítás. Amikor a telepítő megkérdezi hogy akarjuk -e most konfigurálni a szoftvert mondjunk igent, vagy ha inkább elhalasztanánk akkor nemet, ez esetben lehetőség van a `vmware-config.pl` parancs segítségével a szükséges beállítások elvégzésére később is.

A beállítás fontosabb lépései

Mielőtt futtatnánk a beállító szkriptet vessünk egy pillantást a *VMware* által támogatott gazda rendszerek listájára. Ha az általunk futtatott disztribúció szerepel itt akkor könnyű dolgunk lesz, ezeknek kerneljeihez előre fordított modul tartalmaz a *VMware* installációnk így nem kell őket lefordítanunk. Ha a listán nem szereplő rendszeren szándékozunk használni a szoftvert akkor néhány csomagot föl kell telepítenünk, ezek pedig a *gcc*, *make*, és az éppen futó kernelünk fejlécállományai.
Most már minden készen áll a beállító szkript futtatásához, váltsunk rendszergazda felhasználóra, majd indítsuk el a `vmware-config.pl` paranccsal.
Első teendőnk a *VMware* licenc elolvasása és elfogadása. Következő fontos lépés a kernel modulok lefordítása és betöltése. Először a beállító szkript megpróbál előre fordított modult találni a futó kernelhez, ha ez nem sikerül neki akkor lefordítja forrásból, ehhez szükségesek a *gcc*, *make* és a fejlécállományok (header-ek). Ilyenkor megkérdezi a *gcc* helyét ha azt automatikusan a `CC` környezeti változóból nem tudja meg, csomagból telepített *gcc* esetén ezzel elvileg nem lehet gond. A másik kérdés a fejlécállományok helye, ha nem jó itt az alapbeállítás akkor változtassuk meg, de nekem eddig mindig megtalálta a megfelelő fejlécállományokat.
Most néhány kérdést a hálózattal kapcsolatban fog feltenni, elsőként hogy szeretnénk -e egyáltalán hálózati támogatást virtuális gépeink számára, válaszoljunk igent. Ezt követően automatikusan beállít egy *hálózati hidat (bridged network)*. Következő kérdésben megkérdezi hogy szeretnénk-e *NAT hálózatot (NAT networking)* beállítani, itt is igent válaszoljunk, és hagyjuk azt is, hogy találjon egy *használaton kívüli belső hálózati címet (unused privat network)*. Több *NAT* hálózat beállítása általában nem szükséges, válaszoljunk nemmel.
Most a *gazdagépen belüli hálózat (host-only network)* beállítása következik, ezekre a kérdésekre is válaszoljunk hasonlóan mint a *NAT* hálózattal kapcsolatos kérdésekre. Ezután még lefordít egy modult – ebbe nincsen beleszólásunk –, majd vissza adja a parancsjelet (prompt).
Ha valamit elrontottunk a konfigurálás során vagy megváltoztatnánk bizonyos beállításokat futtassuk újra a *vmware-config.pl* szkriptet.

Virtuális gépek létrehozása

Telepítettük, beállítottuk, nincs más hátra, próbáljuk ki a *VMware*-t. Jelentkezzünk be grafikus felületen majd parancssorba gépeljük be a `vmware` parancsot, ekkor megjelenik a szoftver grafikus felülete.
Első teendőnk hogy létrehozzunk egy virtuális gépet, ezt a *New virtual machine* gombra kattintva tehetjük meg. Egy varázsló jelenik meg előttünk ennek a segítségével „összeszerelhetjük" virtuális gépünk. Az első lépésben aki egy egyszerű általános kiépítésű virtuális gépet szeretne magának az válassza a *Typical* menüpontot, aki inkább beleszólna a részletekbe az a *Custom*-ot jelölje be. Következő lépésben a varázsló megkérdezi hogy milyen vendégrendszert szeretnénk telepíteni, én *Linuxot* választottam majd a legördülő menüből az *Other Linux 2.6.x kernel*-t. Aki legelső lépésnél a *Customot* jelölte be az kap még egy kérdést a virtuális gép típusáról, mindenképp a *New – Workstation 5*-öt érdemes választani, ha csak nem akarjuk a szoftver régebbi verzióját használni. Ezután a virtuális gép helyét kell megadnunk. Ez követően a hálózati kártya beállítására kerül sor, egyenlőre az utolsó menüpontot jelöljük be, később részletesen kitérünk ezekre a lehetőségekre. Aki az elején a *Custumot* választotta annak még be kell állítani a virtuális gép rendelkezésére bocsátott memóriaterület nagyságát. Legutolsó lépésként a virtuális gép merevlemezének konfigurálása a feladat, aki a *Custom* ágon indult el annak itt is több lehetősége lesz mint aki a *Typicalt* választotta.
Előbbi felhasználók megválaszthatják a virtuális merevlemez interfészét (*SCSI* vagy *ATA*) de arra is van lehetőségük hogy fizikai lemezt használjanak, ezt



■ *2. ábra* A Virtual Machine Settings ablak

azonban csak tapasztalt felhasználóknak javasolja a *VMware*. Utóbbinak akkor van nagy jelentősége ha a merevlemezünkre már korábban telepített operációs rendszert szeretnénk elindítani virtuális gépünkön. Ha a *Specify Disc Capacity*-hoz érünk már nincs sok hátra, ne kapkodjuk el a döntést gondoljuk át mekkora merevlemezt hozunk létre nehogy kifogyunk a rendelkezésre álló helyből ugyanis ilyenkor elég macerás megnövelni a lemez méretét. Itt szeretném két dologra felhívni az olvasó figyelmét, elsőként az *Allocate all disc space now* jelölőnégyzetre. Ha ezt bejelöljük akkor az egész merevlemeznek lefoglaljuk a helyet előre, ha nem tesszük akkor menet közben nő a merevlemez. Ezzel sebességnövekedés érhető el a tárterület rovására, a szerző az olvasóra bízza a döntést. Másik amire fölhívnám a figyelmet az a Cancel gombtól balra elhelyezkedő *Advenced* gombra, ez csak az utolsó lépésnél érhető el és akkor is csak azoknak akik az elején a *Custom* ágat választották.
A beállítások végeztével nyomjuk meg a *Finish* gombot. A virtuális gépünk megjelenik a *Favorites* mezőben, kész vagyunk kezdhetjük a vendégrendszer telepítését. Arra is van lehetőség hogy több virtuális gépet létrehozzunk az előzőek ismétlésével.
Mielőtt elindítanánk virtuális gépünket vessünk egy pillantást a hardverkiépítésre, és a virtuális gép beállításaira. Ehhez jelöljük ki a *Favorits* ablakban virtuális gépünket majd kattintsunk az *Edit virtual machine settings* gombra. Ekkor a 2. *ábrán* látható ablak jelenik meg. Itt a *Hardware* fülön tudjuk a „beszerelt" hardvereket finom hangolni, új hardvereket „berakni" vagy eltávolítani. Az *Options* fülön több apróságot lehet beállítani, én egyenlőre egyet emelnék ki a Shared folders menüpontot. Itt tehetjük elérhetővé a virtuális gépeink számára a *gazdarendszerünk állományrendszerének (file system)* egyes részeit. Ezek a *UNIX* típusú vendégrendszereken a */mnt* alá *becsatolva (mount)* jelennek meg, *Windowsnál* pedig a *Sajátgépben (My Computer)* önálló meghajtóként látható, a *VMware Tools* telepítése után (erről még lesz szó).
Ha *Linuxot* szándékozunk telepíteni, nem árt egy apróságot átállítani. Nyissuk meg a *Preferences* ablakot (*Ctrl+P*) az Edit menüből majd kattintsunk a *Hot keys* panelra. Itt tudjuk beállítani hogy milyen billentyűkombinációval lehet teljes képernyős módból, ablakba ültetett módba állítani a szoftvert. Ennek azért van jelentősége mert *Linuxnál* az *Alt+Ctrl+Fx* billentyűk együttes lenyomásával lehet grafikus felületről karakteresre váltani, ha viszont lenyomjuk az *Alt+Crtl* kombinációt rögtön visszavált ablakba ültetett módba a *VMware*, és az *Fx* billentyű lenyomása után nem a vendégrendszer x. konzoljára, hanem a gazda rendszer x. konzoljára vált a képernyő. Hogy ezt elkerüljük állítsuk át Hot Key Combination-t *Ctl-Shift-Alt*-ra majd nyomjuk meg az OK gombot.

Vendégrendszer telepítése

Most már minden készen áll a telepítésre. Helyezzük a (de ne csatoljuk föl) telepíteni kívánt operációs rendszer lemezét *CD/DVD* meghajtónkba. Ha iso lemezképből (cd image) szeretnénk telepíteni akkor a már bemutatott *Virtual machine settings* ablakban a *CD-ROM* beállításainál lehet átállítani a *VMware*-t hogy ne fizikai meghajtót hanem a lemezképet használja a *CD/DVD* meghajtó emulációjánál. Most nyomjuk meg a *Power On* gombot (a menüsor alatt), majd a *Full Screen* gombot, ekkor elindul virtuális gépünk. Most ismerős képpel kell találkozni, a *VMware* ugyanis emulálja a szabványos bootfolyamatot és egy szabványos *PC BIOS*-t (*PhoenixBIOS™ 4.0 Release 6 with VESA BIOS*) is. Amikor megjelenik virtuális gépünk képernyőjén a nagy *VMware* logó nyomjuk meg az *F2* gombot, ezzel belépünk a virtuális gép *BIOS*-ába. Lépjünk a *Boot* fülre majd a *CD-ROM Drive*-ot állítsuk első helyre (segítség az *Item SpecificHelp* felirat alatt), így először a *CD-ROM*-ról próbál operációs rendszert indítani virtuális gépünk. Innentől minden úgy zajlik mintha fizikailag létező gépre telepítenénk operációs rendszert. Egy operációs rendszer általában nem tartalmazza a számítógépbe szerelt összes hardver működtetéséhez szükséges meghajtóprogramokat (driver), ezért telepítés után ezt a hardver gyártója által mellékelt lemezről kell fölrakni. Mivel a *VMware* néhány hardvert – például *VGA* kártya, alaplapi lapkakészlet, hálózati csatolót, stb. – emulál ezért ez a virtuális gépeknél sincsen másképp, a „számítógép gyártója" itt is adja a meghajtó lemezt *MS Windows*, *Linux*, *FreeBSD* és *Novell NetWare* operációs rendszerekhez; a kicsomagolt *vmware-distrib* könyvtáron belül a *lib/isoimages/* könyvtárban találhatóak *iso* lemezkép formátumban. Ezek a meghajtó programok hardverek vezérlésén kívül sok kényelmi funkciót nyújtanak például a fentebb említett *Shared Folders* sem működik nélkülük. Ezeknek a meghajtó programoknak az összefoglaló neve a *VMware Tools*, most röviden linuxos és windowsos változat telepítését mutatom be.
Linux esetén ismét *rpm* és *tgz* formában kapjuk a meghajtóprogramot. Itt is érdemes meglátogatni a *VMware* honlapját és utánanézni a támogatott vendégrendszereknek, ugyanis ezekhez van *rpm* telepítője és előre fordított kernelmodulja a *VMware Tools*-nak. Miután felcsatoltuk a *linux.iso* nevű lemezt vendégrendszerünkön váltsunk a *CD* csatolási könyvtárába majd rendszergazdaként adjuk ki az

```
rpm -Uhv VmwareTools-5.0.0-
➥13124.i386.rpm
```

parancsot. Ekkor megszokott módon települ az *rpm* csomagból a program. Most konfigurálni kell a *VMwareTools*-t ezt a

```
vmware-config-tools.pl
```

paranccsal lehet. Ha támogatott rendszerünk van akkor valószínűleg nem kell semmit tennünk, a konfiguráló szkript mindent megold magától, a meghajtók települése után indítsuk újra virtuális gépünket; ezzel készen is vagyunk.
Aki *tgz*-ből telepít ismét nehezebb fába vágja a fejszéjét. Itt is előfeltétel a *tar*, *gzip*, *gcc*, *make* programok és a futó kernelhez tartozó fejlécek megléte a virtuális gép operációs rendszerén. Elsőnek jelentkezzünk be rendszergazda felhasználóként vendég-

rendszerünkbe majd tegyük be a ***linux.iso*** lemezt és csatoljuk föl. A lemezről a ***VMwareTools-5.0.0-13124.tar.gz*** nevű fájlt másoljuk saját könyvtárunkba, majd lépjünk át ide. A

```
tar -vzxf  VmwareTools-5.0.0-
➥13124.tar.gz
```

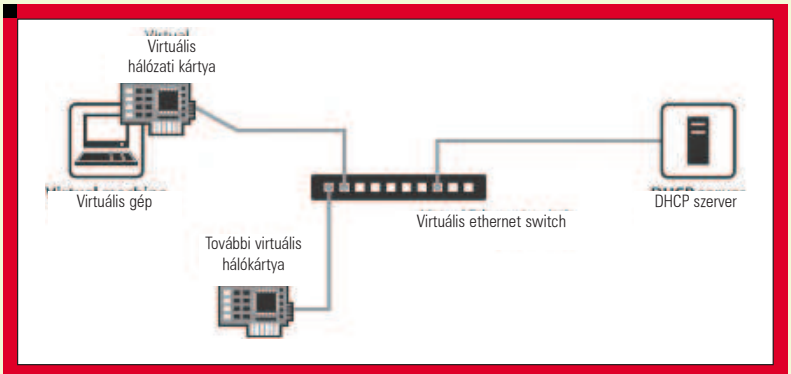
paranccsal csomagoljuk ki, lépjünk be a ***vmware-tools-distrib*** könyvtárba, majd adjuk ki a

```
./vmware-install.pl
```

parancsot a telepítés megkezdéséhez. Először néhány fájl telepítési helyére kérdez rá ezeknél megfelel az alapértelmezett. Amikor ahhoz a kérdéshez ér, hogy futtassa-e a ***/usr/bin/vmware-config-tools.pl*** programot, döntsünk belátásunk szerint. A konfigurálásra később is van lehetőség.
Ha igent válaszoltunk vagy kiadtuk a `vmware-config-tools.pl` parancsot, akkor először megpróbál találni egy előrefordított kernelmodult de ezt valószínűleg nem fog sikerülni. Ekkor rákérdez hogy megpróbáljon-e forrásból fordítani, válaszoljunk igennel. Valószínűleg a *gcc* és a fejlécek helyét is jól állítja be magától, ha esetleg nem, adjuk meg a helyes elérési utat. Ezután lefordítja a szükséges modulokat, majd megkérdezi a csatlakoztatott monitor felbontását, itt a valós adatot adjuk meg ugyanis a monitort nem emulálja a *VMware*. Kicsit elkezd „villogni" a képernyő majd, véget ér a konfigurációs szkript, végeztünk a beállításokkal, indítsuk újra virtuális gépünket.
Ha *MS Windows*-t választjuk vendégrendszernek egyszerű dolgunk lesz, csak a ***windows.iso*** nevű lemezt kell betennünk virtuális gépünkbe, majd dupla klikk a lemez ikonján a ***Sajátgépben*** és indul a telepítés. Szokásos módon az ***Install Shild Wizard*** segít ebben, nincs egyéb dolgunk mint a ***Next*** gombot nyomni végig, majd a telepítés végeztével újraindítani a virtuális gépet.

VMware hálózat

Mint már láttuk a *VMware* sokféle hardvert képes emulálni, nincs ez másképp a hálózati kártyákkal sem, ebben a részben érintőlegesen tárgyaljuk a *VMware* hálózat emulációval kapcsolatos szolgáltatásainak elméletét és gyakorlatát. Sajnos itt nem lesz módunk alaposan megismerkedni



3. ábra A Host-only hálózat felépítése

ezekkel a lehetőségekkel, inkább a *VMware* egyszerűbb általánosabb képességei bemutatására kerül sor. Hálózati kapcsolatot úgy biztosíthatunk virtuális gépünk számára, hogy „beszerelünk" egy hálózati kártyát. Ezt a már korábban bemutatott ***Virtual machine settings*** ablakban tehetjük meg. Kattintsunk az ***Add...*** gombra majd a ***Hardware Types*** menüből válasszuk ki az ***Ethernet Adapter***-t és nyomjunk ***Next***-et. Négy választási lehetőség tárul elénk, itt válasszuk ki azt a hálózatot amire csatlakozni kívánunk majd nyomjunk meg a ***Finish*** gombot. Ha további hálózati csatolót szeretnénk virtuális gépünkbe helyezni ismételjük meg az előző lépéseket. Az utolsó lapon négy fajta hálózat közül választhatunk:

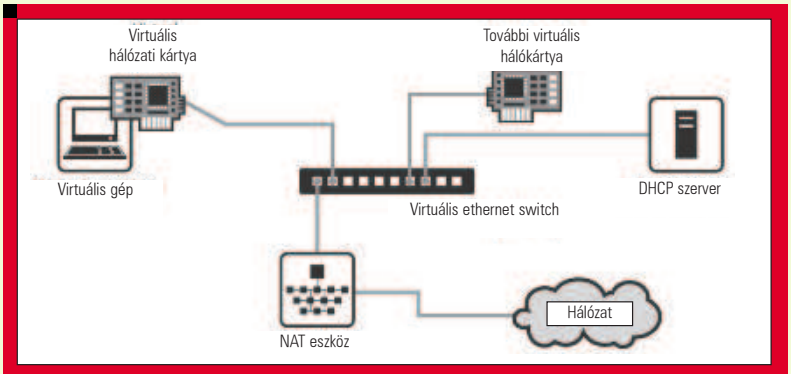
- ***Bridged***: közvetlenül a hálózatra kapcsolva
- ***NAT***: a gazdarendszer IP-címének közös használata
- ***Host-only***: belső hálózat a gazdarendszer és a virtuális gép között
- ***Custom***: egyéb

Bridged

Virtuális gépünk hálózati kártyáját egy ***híd (bridge)*** segítségével közvetlenül ahhoz a fizikai hálózathoz köti amelyikre gazdagépünk csatlakozik. Ha több hálózati kártya van számítógépünkben akkor létrehozhatunk még egy hidat ami a második csatolón keresztül köti virtuális gépünket a fizikai hálózathoz. Mivel virtuális gépünk fizikai hálózat részévé válik ezért a virtuális hálózati csatoló *IP* címét a fizikai hálózaton használt címtartományból kell választanunk. Virtuális gépünk külső fizikai gépekről is „látható" – azaz tudjuk pingelni. Mint minden hálózati kártyának virtuális csatolónknak is van ***fizikai címe (MAC address)***. Ezek a címek a gyártás során kerülnek kiosztásra, így elvileg minden fizikai cím egyedi. Mivel a *VMware* virtuális kártyái nem léteznek ezért ezek nem is kaphattak gyártás közben ilyen címet, amikor hozzáadunk egy virtuális kártyát egy virtuális géphez akkor a *VMware* véletlenszerűen hozzárendel egy fizikai címet, így előfordulhat hogy címegyezés lép föl a hálózatba kötött két hálózati csatoló között, bár ennek kicsi az esélye. Ha ilyet észlelnénk természetesen van lehetőség a virtuális kártya fizikai címének megváltoztatására is.

Host-only

Olyan hálózat amely egy gazdagépen belül létezik, teljesen elszigetelt a fizikai hálózattól, csak a gazdagépen futó virtuális gépek és a gazdagép csatlakozhatnak hozzá. A virtuális gépek ehhez a hálózathoz is virtuális hálózati kártyával kapcsolódnak. Ha bármelyik virtuális gépünkbe „beszerelünk" egy ***host-only*** hálózati kártyát akkor gazdagépünkben is megjelenik egy virtuális hálózati kártya, ennek segítségével kapcsolódik a gazdagép a virtuális hálózatba (futtassuk rendszergazdaként gazdarendszerünkön az `ifconfig` parancsot a hálózati kártyák kilistázásához).
Mivel ennek a hálózatnak nincs köze semmilyen fizikai hálózathoz ezért a virtuális hálózatunkon fut egy *DHCP* kiszolgáló (a *VMware* gondoskodik róla) ettől kérjük az *IP* címet és egyéb hálózati információkat. A ***host-only*** hálózat működését a *3. ábra* mutatja. Természetesen a virtuális switchbe beköthetünk több virtuális gépet is.



4. ábra A NAT hálózat felépítése

NAT

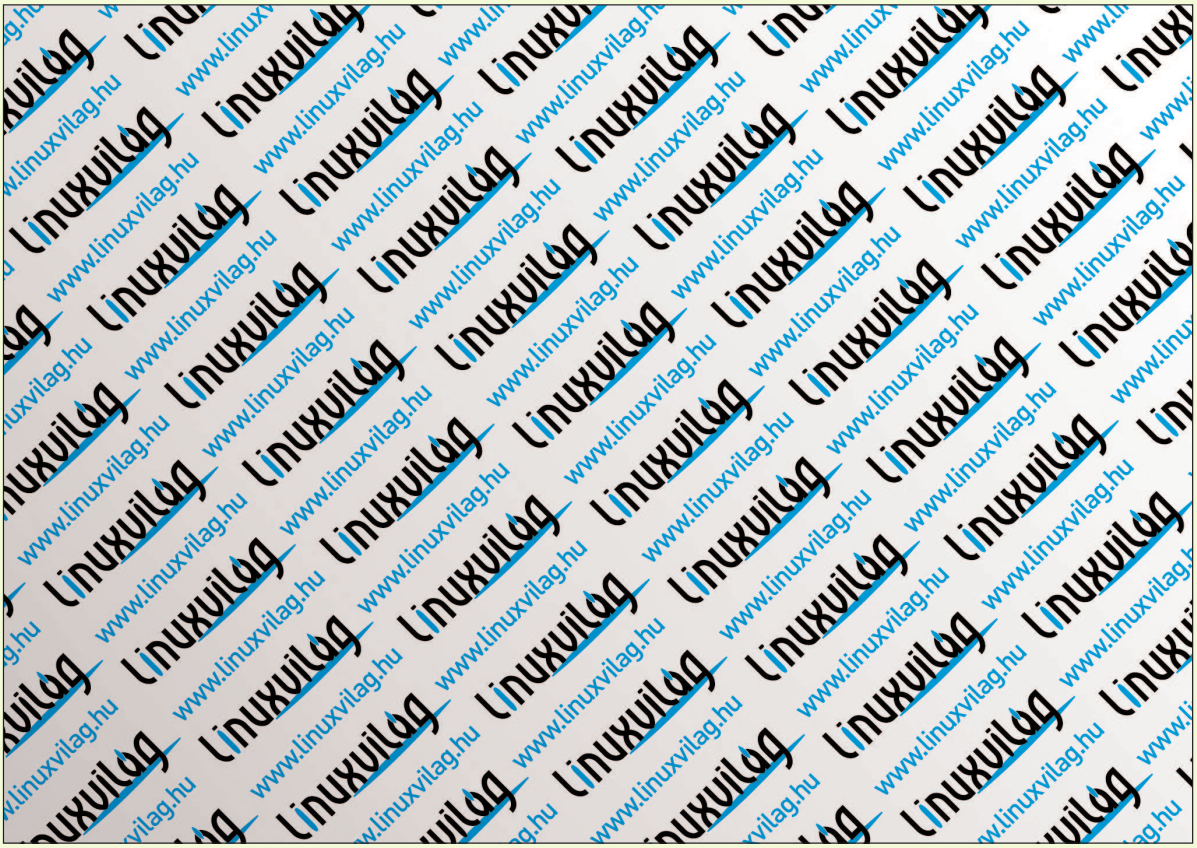
Ennek a hálózatnak a kiépítését az *5. ábra* szemlélteti. Sok mindenben hasonlít a ***host-only*** hálózat felépítéséhez. Itt is egy virtuális switch-be vannak bekötve a virtuális gépek és egy virtuális hálózati kártyán keresztül a gazdagép, itt is fut *DHCP* szerver a hálózaton. A különbség csupán annyi, hogy egy *NAT* doboz is be van kötve a switch-be, ez végzi a címfordítást a fizikai és a virtuális hálózat közt. Erre a hálózat típusra akkor lehet szükség ha virtuális gépeinkről szeretnénk elérni a fizikai hálózatot de csak egy *IP* cím áll rendelkezésünkre, például ilyen a betárcsázós internet. A *NAT* dobozoknak köszönhetően a virtuális hálózatról nem kerülhet ki olyan csomag amely a belső virtuális hálózatban használt címtartományból származó *IP* címmel rendelkezik, természetesen ez érvényes a *DHCP* kérésekre és válaszokra is, ezért nem fordulhat elő hogy fizikai hálózatot elöntik a virtuális hálózatból jövő válaszok. A virtuális gépeknél a *DNS* szerver címét a *NAT* doboz címére állítsuk, ugyanis az *DNS Proxy*-ként is üzemel, továbbítja a *DNS* kéréseket. A *NAT* doboz mögött lévő gépek kezdeményezhetnek hálózati kapcsolatot és kéréseikre választ is kaphatnak, viszont külső gépek nem csatlakozhatnak a virtuális hálózat gépeihez. Ez utóbbi azért nem szent írás, a ***Port Redirection*** nevű szolgáltatással lehet ezen segíteni, de ez nem témája ennek az írásnak.

Custom

Ezzel a lehetőséggel csak említés szinten foglalkozunk. Főleg akkor szokás használni ha egy komplett hálózatot szeretnénk emulálni. Akit érdekel bővebben az olvassa át a ***User's Manual Custom Networking Configurations*** lapját.

A Snapshot és a Clone funkciók

Szoftverrendszerek teszteléses során gyakran előfordul hogy egy kisebb hiba kijavítása közben valamit elvétünk ezért nagyobb galiba keveredik és újra kell telepítenünk az egész rendszert, ezért időnk nagy részét nem hasznos munkával hanem tesztkörnyezetünk



karbantartásával töltjük. Előbbi hatványozottan igaz ha egy komplett szerverrendszert emulálunk.
Hogy tesztkörnyezetünket minél hamarabb visszaállítható legyen a *VMware*-be beépítették a ***Snapshot (pillanatkép)*** funkciót. Ennek segítségével bármikor készíthetünk virtuális gépünkről pillanatképet, majd ha szükséges visszatölthetjük.
Pillanatképet a menüsor alatt megtalálható ***Snapshot*** gombbal készíthetünk futó és kikapcsolt (***Suspend*** is ide értendő) állapotban lévő virtuális gépről. Miután megnyomtuk a gombot megjelenik a ***Take snapshot*** ablak., itt tudjuk elnevezni a pillanatképet és megjegyzést fűzni hozzá. Az ***OK*** gomb lenyomása után elkezdődik a pillanatkép készítése – ez körülbelül 10 másodpercet igényel –, de ebből semmit sem észlelünk, a háttérben zajlik minden. Egy pillanatkép a memória állapotát, a merevlemezek állapotát és a virtuális gép beállításait tartalmazza. Erőforrás takarékossági okokból nem menti le az egész lemezt egy pillanatkép csak a ***You are here***-hez képest történt változásokat.
Ennek van hátránya is ugyanis így minden változást a virtuális gépek működése közben kerülnek a merevlemezre ezzel tovább lassítva az amúgy is lassú lemezelérést, épp ezért a pillanatkép szolgáltatást le lehet tiltani a ***Virtual Machine Settings*** ablak ***Options*** fülén a ***Snapshot*** menüben.
Természetesen több pillanatképet is készíthetünk. Pillanatképeinket a ***Snapshot Manager*** nevű ablakkal kezelhetjük. Ennek az ablaknak a segítségével pillanatképeinket fába szervezhetjük. Ennek akkor vehetjük nagy hasznát ha egy szoftver különböző verzióit használva próbáljuk megvalósítani ugyan azt a szolgáltatást.
Mivel a kép tartalmazza a memória állapotát is ezért miután visszalépünk egy pillanatképre a futó folyamatok ott folytatódnak ahol a kép készítésének pillanatában tartottak, ezért lehetőleg akkor készítsünk pillanatképeket amikor virtuális gépük nem folytat hálózati kommunikációt.
Pillanatkép visszaállításához a ***Snapshot Manager***-ben kell kijelölni a kívánt állapotot majd az ablak alján található ***Go To*** gombot megnyomni.

Miért használjunk VMware-t?

- Több operációs rendszer fejlesztése és tesztelése egy számítógépen
- Virtuális gépek összekapcsolása több szintű konfigurációk szimulálására és tesztelésére
- Pillanatképek és hibakeresési támogatás használata a könnyű teszteléshez
- Teszt környezet arhiválása fájl szervereken ahonnan egyszerűen lehet visszaállítani
- Asztali és szerverkörnyezet tesztelése virtuális gépeken mielőtt valós környezetben való használat előtt
- Több szintű alkalmazások, frissítések, foltok tesztelése egy számítógépen
- Régi alkalmazáskörnyezet tárolása virtuális gépen ezzel segítve az operációs rendszer váltást
- Tantermi anyagok előre csomagolása
- Elérhetővé tenni a tanulók számára hogy több operációs rendszeren és különböző alkalmazásokon kísérletezzenek, egy biztonságos elszigetelt környezetben
- Komplex és több szintű alkalmazások bemutatása egy laptopon

A ***Clone*** funkció akkor jön jól ha egy virtuális gépet sokszorosítani szeretnénk. Erre akkor lehet szükség ha mondjuk két ***Debian Linuxot*** telepítünk két virtuális gépre, egyiket szervernek másikat asztali gépnek szánjuk. A telepítés folyamata csak az alaprendszer telepítése után válik ketté, a ***Clone*** funkciónak köszönhetően az alaprendszer telepítését csak egyszer kell elvégeznünk, ezzel időt takarítunk meg.
Minden kikapcsolt állapotba lévő virtuális gépet és kikapcsolt állapotba készített pillanatképet lehet sokszorosítani, ehhez kattintsunk a ***Clone this virtual machine*** gombra majd nyomjuk meg a ***Next*** gombot. Itt válasszuk ki mit akarunk klónozni majd nyomjunk ***Next***-et. Ezen a lapon a virtuális gép típusát kell kiválasztanunk. ***Linked clone***-t akkor készítsünk ha lesz hozzáférése a klón tulajdonosának az eredeti virtuális géphez, így tárterületet takaríthatunk meg. ***Full clone***-t csak akkor érdemes készíteni ha az eredeti virtuális gép nem lesz elérhető a klón használója számára, ilyen esetben a komplett virtuális gép lemásolódik. Következő lapon válasszuk ki a klón helyét, nyomjunk ***Next***-et majd ***Finish***-t.

Utószó

Ebben a cikkben természetesen csak arra volt lehetőségem, hogy a ***VMware*** alapfunkcióit bemutassam. A ***Miért használjunk VMware Worksationt?***-t című keretes írásban ugyanakkor – továbbra is a teljesség igénye nélkül – kiemeltem a rendszer néhány hasznos tulajdonságát, illetve egyes lehetséges felhasználási területeket.
A ***VMware Inc.*** cégnek a ***VMware Workstation*** nem az egyetlen terméke, bár kétségtelen, hogy ez a legrégebbi termékük, és ez a vállalat zászlóshajója. A cég másik két hasonló terméke a ***VMware GSX*** és ***VMware ESX*** szerver. Ezek is ***PC*** emulátorok, de inkább nagyobb vállalatok számára készültek: nem asztali gépekre, hanem szerverre kell telepíteni őket, a felhasználók pedig hálózaton keresztül használhatják virtuális gépeiket.

Szilágyi Attila (szati1@invitel.hu)
Néhány éve használ Linuxot. Alapvetően minden ezzel a témával kapcsolatos felhasználási terület érdekli és szívesen fogadja bárki kérdést, észrevételét.

KAPCSOLÓDÓ CÍMEK

Microsoft Virtual PC honlapja:
➲ http://www.microsoft.com/windows/virtualpc/default.mspx

A VMware Inc.-ről röviden:
➲ http://www.vmware.com/company/

VMware letöltés:
➲ http://www.vmware.com/download/workstation.html

VMware Dokumentáció:
➲ http://www.vmware.com/support/pubs/ws_pubs.html

QEMU: dobozba zárt univerzum

Iskolánkban nemrég áttértünk az UHU-Linux 1.1-es változatáról az 1.2-esre. Ez egyrészt sok örömteli újdonságot hozott, ám egyben a glibc2.0 támogatás megszűnését is jelentette. Ennek köszönhetően aztán a csak bináris formában rendelkezésre álló Maple 7 programunkat el se lehetett indítani...

Amikor segítséget kértem a *Maplesofttól*, azt a lakonikus választ kaptam, hogy a *Maple 7* program támogatása már megszűnt. („*Please note that Maple 7 is no longer supported by Maplesoft*”.) A program *UHU 1.1* alatt ugyan nagyszerűen működik, viszont pusztán emiatt nem szerettem volna lemondani az annyi újdonságot hozó 1.2-es változatról. Eleinte *chroot*-olt környezettel kísérleteztem, de az operációs rendszer *glibc2.0* helyett *glibc2.1*-re (azaz *LinuxThreads*-ről *Nativ POSIX Thread Library*-re, *NPTL*-re) való átállása olyan mélyreható változásokat hozott, amit nem tudtam egyszerű kézimunkával helyreigazítani. Ekkor a *QEMU*-val próbáltam ki egy olyan lehetőséget, amely tulajdonképpen messze túlmutat az eredeti probléma megoldásán.

A QEMU-ról

Az UHU levelezőlistán kaptam az ötletet, hogy érdemes volna megpróbálkoznom a QEMU-val. Van hivatalos *.uhu* csomag is belőle, de mivel meg szerettem volna őrizni a leírás általánosságot, tehát nem akartam egy adott terjesztéshez kötődni, a cikk megírása során nem ezt használtam. E programnak természetesen vannak kereskedelmi vetélytársai (*Wabi, Virtual PC, VMWare, Serenity Virtual Station, TwoOstwo*), de számomra természetesen fontos szempont volt, hogy szabad szoftvert válasszak – részben a közismert anyagi okok miatt, de azért is, mert ennek lehet jobban a „lelke mélyére nézni”, ami az oktatásban szintén lényeges szempont.

A *QEMU* alkotója *Fabrice Bellard* (a név az ő bejegyzett védjegye), honlapja pedig a *http://fabrice.bellard.free.fr* helyen található. Amint a program neve is sejteti, egy emulátorról van szó: amely az *x86* architektúra mellett néhány más (*ARM*, *PowerPC*, *SPARC*) processzort is képes emulálni. Mindezt dinamikus fordítással, és egészen jó sebességgel.
A felhasználói licenc különböző részekre oszlik. A program lelkét jelentő virtuális processzor törzskönyvtár (*virtual CPU core library*) és a *PC* rendszer-emulátor *LGPL* alá tartozik, míg a felhasználói térben történő (*user mode*) emulátorhasználatra a *GPL* vonatkozik. Ugyanakkor a gyorsítómodul (*kqemu*) szabadalmaztatott termék. Ingyenesen hozzáférhető – ám a karbantartója tudni szeretné, ha valahol közzéteszik. (A modul nevének kezdőbetűje kissé megtévesztő, ugyanis a rendszernek semmi köze a *KDE* környezethez.)
A *QEMU*-nak kétféle futási módja van:

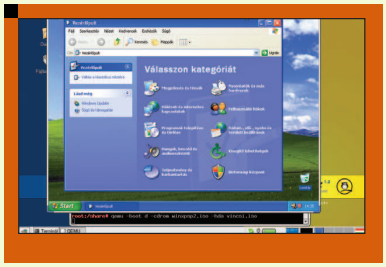
- *Teljes rendszeremuláció*. Ilyenkor a *QEMU* a teljes *PC*-t emulálja, beleértve a processzort és a különböző perifériákat is. Így más operációs rendszerek is futtathatók vele, sőt nyomkövetést is végezhetünk.

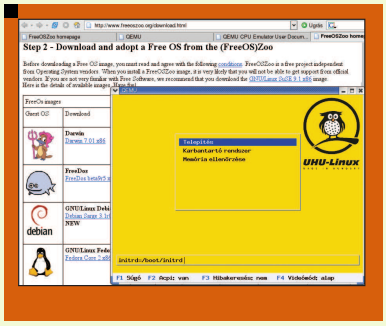
- *Felhasználói térben történő emuláció (csak Linuxon)*. Ilyenkor a *QEMU* a processzoron olyan programot is el tud indítani, amit más processzorhoz fordítottak. Ez segítségére lehet például a *wine* fejlesztőinek és használóinak is.

Számos érdekessége közül az egyik, ami jól jellemzi az emuláció minőségét: akár önmagán belül is elindítható egy másik *QEMU* emulátor. Ezt még a kereskedelmi vetélytársak mindegyike sem tudja produkálni.

A QEMU előkészítése

A cikk megírásakor forráskódban a *qemu-0.7.0* volt elérhető a fenti honlapon, ezt letöltöttem. Néhány kellemetlen órát okozott (meg néhány tanácstalan levélváltást a „haladó *UHU-Linux*” levelezőlistán), hogy sehogy sem akart lefordulni a forráskód – végül kiderült, hogy első próbálkozásra nem jött le a teljes tömörített fájl, csak a nagyobb része. Érdemes erre figyelni a letöltéskor.
A fordításhoz kell a *texi2html* program is, ez készíti el a *qemu-doc.html* és *qemu-tech.html* fájlokat, amit haszonnal lehet forgatni a továbbiakban. Letöltöttem a *kqemu-0.6.2-1* gyorsító modult is. Kibontás után a *kqemu* könyvtárat (tehát nem a tartalmát) bemásolva a *qemu-0.7.0* könyvtárba, szépen lefordult a program. A keletkezett binárisok a */usr/local/bin*-be kerültek a (rootként





végrehajtott) `make install` után: *qemu*-val kezdődnek neveik, és a különböző processzorok neveivel folytatódnak. Kivétel ez alól a *qemu-img*, ami a hivatalos *.uhu* csomagban *qemu-mkcow* néven szerepel: ez nem emulátor-indító, hanem képfájl készítő program.
A gyorsító modul betöltése disztribúciótól függően eltérően történhet.
A kézi megoldás biztosan működik, root felhasználóként kiadva az alábbi parancsot:

```
modprobe kqemu; rm -f /dev/
➥kqemu; mknod /dev/kqemu c 250
➥0; chmod 666 /dev/kqemu
```

Ellenőrzésként:

```
>modinfo kqemu
vermagic:       2.6.9-19 SMP
➥586 gcc-3.3
```

Ez a kézi töltögetés várhatólag néhány hónapon belül egy kulturálisabb megoldást kap az *UHU-Linux* esetében (és bizonyára más disztribúciókban is). Arról, hogy hogyan készül el az a (lehetőleg nagyméretű) fájl, amit a *QEMU* mint merevlemezt vagy *CD-ROM*-ot kezel, majd később ejtek szót részletesebben. Mindenesetre már itt is fontosnak tartom megemlíteni: minden egyes perifériát mi adhatunk meg az emulátornak egy-egy megfelelően előkészített fájl formájában.

Állatkerti séta

Mielőtt nekiláttam az eredeti célkitűzést jelentő programkörnyezet-kialakítási munkának, körülnéztem a *QEMU* honlapján. Találtam egy érdekes hivatkozást: *„Szabad Operációs Rendszerek Állatkertje" (Free Operating System Zoo):* ➲ *http://www.freeoszoo.org*
Számos (szabad) operációs rendszer képfájlja (*.iso* fájlja) van itt összegyűjtve, kifejezetten a *QEMU*-ban való futtatásra. Maga a *QEMU* is többféle rendszeren futhat (gazdagép, *host*) – lehet *Linux*, *MacOs X* vagy *Microsoft Windows*. A dobozolt (vendég, *guest*) rendszerek pedig számtalanul sokan vannak: a nagyobb *Linux* terjesztések, *BeOS*, *BSD* fajták, *FreeDOS*, *Darwin*, *ReactOS* stb.
Az „állatkert" elnevezés amúgy egyszerre utal a tanulásra és a biztonságra: a futtatott „állatfajta" garantáltan nem fog kárt tenni saját, kényelmesen berendezett világunkban. Ha nagyon akarjuk, az állatokat azért szabad etetni. Elérhető ugyanis a *QEMU*-val az is, hogy a ketrecbe be- vagy onna kijusson egy s más. (Ez nekem ugyebár nagy szerencse, mert különben hogyan futtatnám a Maple-t, vagy hogyan használnám az eredményül kapott fájlokat...)
Persze azért az is sejtethető, hogy az ilyen állapotban tartott élőlények nem olyan sebesen mozognak, mint vadon élő társaik... Pontosabban a sebesség erősen függ a rendelkezésre bocsátott erőforrásoktól (mint ahogy a jobb állatkertek is inkább állatparkok, kellően nagy kifutótérrel az állatok számára). Kimondani is szörnyű, de sokan a pingvint is olyan vadállatnak tekintik, amit inkább rács mögül kell nézegetni. Eme különös embercsoport számára a *Microsoft Windows* alól futtatott *QEMU* lehetőséget ad a többféle *Linux* „dobozban" való futtatására.
Az *UHU-Linux* levelezőlistájának archívumából (*http://lists.uhulinux.hu/arc/halado/2005-06/msg00306.html*) az is kiderül, hogy az „ablakos vadállat"-ot is telepíthetjük és futtathatjuk *QEMU*-ból. Az idézett írás nem kevesebbet állít, mint hogy a *Windows* gyorsabban betöltődhet *QEMU* alól, mint „normál módon". A használt rendszert pedig akár egy *DVD*-re is kiírhatjuk.
A *FreeOsZoo* honlapon nemes egyszerűséggel látszik a *QEMU* futtatásának legegyszerűbb módja:

```
qemu -hda guest_image_name.img
➥-boot c -user-net
```

Ebben a gyűjteményben voltak jó példák, a számomra megoldást jelentő képfájlt azonban – *Szilágyi Szilveszter* ötletét megfogadva – mégis máshonnan szereztem be: a *Damn Small Linux live CD*-jét választottam, 2.4-es kernellel (és persze glibc2.0-val). Kísérleteztem az 1.1-es *UHU*-val is, de ez valahogy ágyúval verébre való lövöldözésnek tűnt, oly nagynak ígérkezett a képfájl, és oly lassúnak a telepítés. Azért az szívdobogtató látvány volt, amikor a

```
qemu -boot d -hda hda.iso -hdb
➥hdb.iso -cdrom uhu1.1.iso
```

parancs hatására először láttam meg *UHU 1.2*-t futtató gépem „dobozában" a régi világból idecsöppent 1.1-es *UHU-Linux* induló telepítőképernyőjét.

A qemu-img, a képfájl-előállító

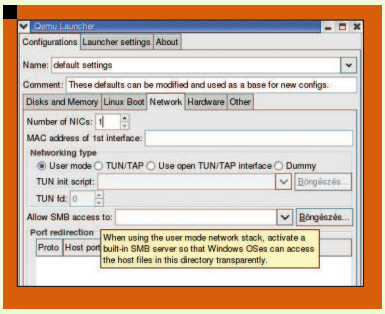
Kiadva a `qemu-img` parancsot, segítséget kapunk a *„QEMU disk image utility"* használatáról.
Egy példa:

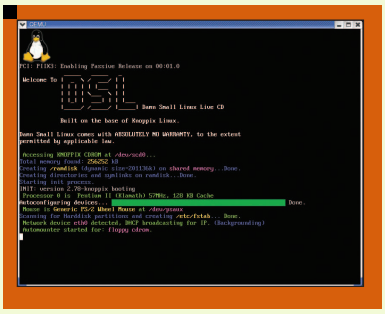
```
qemu-img create hda.iso 100M
```

elkészít egy 100 MB-os image fájlt, ami először üres. (Ne felejtsük le a sor végi `M` betűt!) A típusa többféle lehet és ez természetesen erősen meghatározza a méretet is. Választhatunk tehát `vpc`, `bochs`, `dmg`, `cloop`, `vmdk`, `qcow`, `cow` vagy `raw` formátumot.
Bár a mérete ennek a legnagyobb, sok szempontból az alapértelmezett `raw` az ideális, amelyet aztán a

```
mount -o loop
```

paranccsal könnyedén tudunk fel és lecsatolni, illetve esetleg beletenni valami „eleséget az állatnak".
A `cow` a *copy-on-write* rövidítése, és akkor használható jól, amikor egy csak olvasható képfájlból indulunk ki (*base_image*), és valahol tárolni szeret-



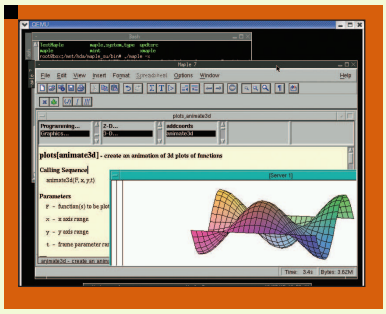


nénk a módosításokat. Gondoljunk csak bele, micsoda nagyszerű lehetőség ez live CD-k testre szabására, megszelídítésére! Erre van kihegyezve a *QEMU* `-snapshot` módja – ennek használatakor minden képfájl írásvédett, és minden változás a */tmp*-ben őrződik. Ezek a változtatások azonban visszaírhatók a képfájlra (*CTRL+Alt+s* paranccsal). A program kezel titkosított vagy tömörített képmásfájlokat is (`qcow` formátum), amiknek a mérete „belülről” (az emulátorból) nézve például 4GB, míg „kívülről” (a gazdagépen mint fájlt tekintve) igazából csak annyi, amennyit a „benti winchesterre” felmásoltunk.

Kiadva a `qemu` parancsot, hosszú listát kapunk a paraméterekről.
Minden megadható, ami csak az ember eszébe juthat: perifériák, nyelv, idő, memória, hálózat, képernyő, grafikus mód, betöltési paraméterek, kernel, initrd. Van ezen kívül számos profi lehetőség, mint például a *gdb* port nyomkövetéshez, eszközök átirányítása... De nem kell kétségbe esni, pilótavizsga nélkül is jól használható mindez, mert jól vannak beállítva az alapértelmezések.
Akit mélyebben érdekel a kérdéskör, annak javaslom a `qemu-launcher` letöltését. Ez egy nagy tudású, *Gtk*, *Gtk::GladeXML* és *Gnome2 PERL* modulokat használó grafikus felület. ***(UHU 1.2 alatt történő elindításához tehát kellenek a perl-gnome2*.uhu csomagok.)*** Aki nem akarja használni, annak is megér egy pillantást, hogy mi minden állítható be (parancssorból is persze) a *QEMU*-ban.

A QEMU bevetés közben

No de térjünk vissza az eredeti kérdéshez: hogyan is fogjuk futtatni a *Maple*-t? Egy olyan kis méretű operációs rendszert kerestem, ami grafikus felületet is ad.
Az internetről is letölthető a ***Damn Small Linux .iso*** fájlja, ami elfér egy névjegykártya méretű CD-n (50MB).



Volt nálam egy ilyen még a tavalyi PHP-konferenciáról, erről elkészítettem az .iso fájlt `dd` paranccsal (`dd if=/dev/cdrom of=dsl.iso`), majd (a fenti `aqemu-img` paranccsal legyártott ***hda.iso*** segítségével) kiadtam a várva várt

```
qemu -boot d -hda hda.iso
➥-cdrom dsl.iso -m 256 -k hu
➥-user-net
```

parancsot.
512 MB RAM van a gépemben, ebből 256-ot szántam az emulátornak (`-m` kapcsoló). Ez nem ment első nekifutásra, de nagyon értelmes és használható hibaüzenetet kaptam, amiből kiderült, milyen paranccsal tudom elérhetővé tenni ezt a nagy memóriaterületet:

```
umount /dev/shm
mount -t tmpfs -o size=272m
➥none /dev/shm
```

A ***QEMU*** annak függvényében állítja be az emulált processzor sebességét, hogy éppen mennyi erőforrás szabad. Ha nem futott más program, 790 MHz is „összejött”, viszont egy ***GIMP*** futtatása mellett (amivel a képernyőképeket begyűjtöttem) néha leesett 20 MHz-re is a sebesség (ami érdekes módon egészen jól használhatónak bizonyult parancssorban).
Aki mélyebben szeretné beleásni magát a ***QEMU*** világába, annak ideális eszköz a ***QEMU monitor*** (***CTRL+Alt+2***), amivel kifinomult módon vezérelhető az emulátor. Használható például kivehető eszközök be- és kicsatolására, naplózására, a ***Virtuális Gép (VM)*** befagyasztására, ami ily módon elmenthető vagy helyreállítható egy korábbi állapotból, valamint egy-egy állapot külső nyomkövetővel való ***VM***-böngészésre. Ebből az állapotból a ***CTRL+Alt+1*** kombinációval lehet visszaváltani a normál nézetre.

Van még egy ***CTRL+Alt+3*** képernyő is, de egyelőre maradjon meglepetés, hogy ennek mi a tartalma.

A vadállat etetése

Először még megformázatlanul vár bennünket a ***hda***-nak szánt ***hda.iso***. Az egeret a ***QEMU*** felületére vihetjük, és rákattinthatunk. Ekkor az egér már a „ketrecben” mozog, ahonnan aztán ***CTRL+Alt*** kombinációval csalogathatjuk ki újra (ez olvasható is az ablak fejlécén, nehogy elfelejtse valaki).
CTRL+Alt+F kombinációval lehet teljes képernyőt kérni (vagy a QEMU indításakor a `-full-screen` kapcsolóval).
Kérjünk egy terminálablakot a „ketrecben”, lehetőleg root-ként. Majd:

```
mke2fs /dev/hda
```

paranccsal megformázható (és esetleg a `tune2fs -j -c0 /dev/hda` segítségével ez ***ext3*** típusúvá is alakítható) a ***hda***-ként jelzett munkaterületünk. Tévedés ne essék: ez ***nem*** az eredeti gép ***/dev/hda***-ja (még akkor sem, ha az `mke2fs` történetesen arról panaszkodik, hogy ugyan már mi ütött belénk, hogy egy teljes merevlemezt szeretnénk megformázni, nem pedig csak egy partíciót?!). Előzőleg önmagunk megnyugtatásaként ki is próbálhatjuk például a `cfdisk` segítségével, hogy valóban akkora-e az a bizonyos ***hda***, amekkorát szerettünk volna.
Ezek után próbaként felcsatolható a frissen formázott félig virtuális merevlemezünk egy egyszerű `mount /mnt/hda` paranccsal (a ***/etc/fstab***-ból már tudja a ***Damn Small Linux***, hogy ezt hová kell csatolni). Adjunk ki egy `touch ittjartam` parancsot, hogy később akárki láthassa, hogy valóban ez az a terület, amihez hozzáférünk innen is, onnan is.
Első körben csak ennyi történt a „ketrecben”. Szépen, kulturáltan lehet kikapcsolást kérni a grafikus felületről – azaz a ***QEMU*** egyelőre befejezheti futását. (Ez elérhető a ***CTRL+Alt+x*** paranccsal is.) Ezek után beadjuk az állatnak az eledelt.
Felcsatolva eredeti operációs rendszerünk valamelyik könyvtárába a ***hda.iso*** fájlt (`-o loop` kapcsolóval) – és feltéve persze, hogy az kellően nagy méretű – belemásolható a ***maple_su (maple single user)*** könyvtár, ahol mindaz megtalálható, amire szükségünk lesz. (Remélhetőleg látszik az ***ittjartam*** fájl is, ami most már törölhető.)
Ezután `umount`-tal lecsatolva a ***hda.iso***-t, indulhat a végjáték. Ismét adjuk ki a

```
qemu -boot d -hda hda.iso
➥-cdrom dsl.iso -m 256 -k hu
➥-user-net
```

parancsot (vagy inkább írjunk egy kis parancsfájlt, amiben ezt eltároljuk).
A grafikus felületen root-ként felcsatolhatjuk a ***hda***-nkat (mint az előbb).

S lám, a vadállatból kezes bárány vált...

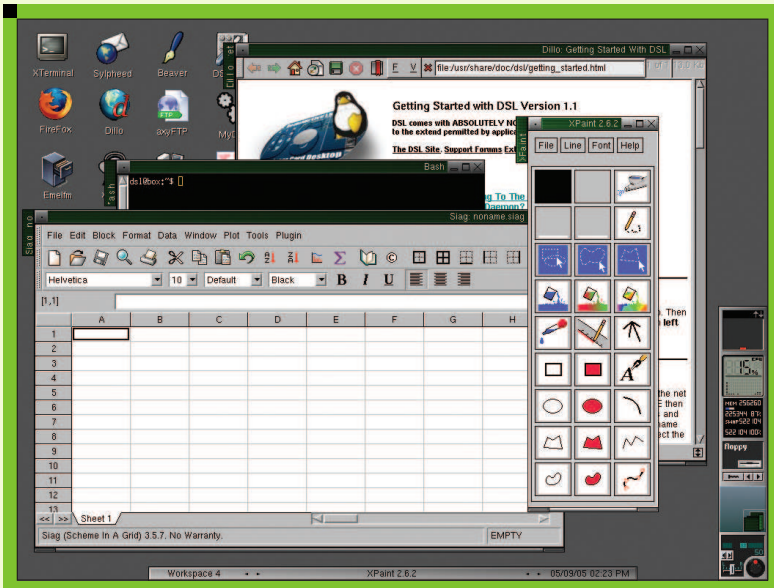
A `/mnt/hda/maple_su/bin/maple` paranccsal karakteres felületet kapunk, ami mondjuk a 888! pontos kiszámolására, a ? első 1000 jegyének kiírására vagy egy harmadfokú egyenlet analitikus megoldására teljesen elegendő.
Az `-x` kapcsolóval azonban elénk tárul a csodálatos világ háromdimenziós (akár mozgó, `animate3d`) ábrákkal. Egy-egy szép matematika dolgozat számára el is menthető, exportálható a kimeneti képernyő, vagy annak egy-egy ábrája (sokféle formátumban), amit szintén a ***hda*** területünkön tárolhatunk, és ha elhagytuk a ketrecet, akkor a jól megszokott operációs rendszerünkben használhatjuk `-o loop` kapcsolóval felcsatolva a ***hda.iso*** fájlt.
Egyetlen képfájlban is konzerválható a ***QEMU***-ban futtatandó rendszer: a 2005. júliusi ***Linuxvilág „Készítsünk Live CD-t!”*** című cikkének segítségével (38. o) tetszőleges kiindulási ***Live CD***-ből elkészíthető egy saját (például ***Maple***-t is tartalmazó) ***.iso*** fájl.
Ily módon ezt az alkalmazást a későbbiekben is jól tudja majd használni bárki, akinek szüksége lesz rá – még akkor is, ha már nem is emlékszik senki arra, hogy valaha használtunk ***glibc2.0***-t.
Sok sikert a ***QEMU*** megismeréséhez!

Szabó Zoltán
Három gyermekével és feleségével Pannonhalmán él. Tíz éve kísérletezik a Linuxszal. Matematikát és informatikát tanít, diákotthonban keseríti a rábízottak életét. Szívügye a PHP és a PostgreSQL.
(szz@freemail.hu)

Kisméretű terjesztések

Egy kölcsönkért gépet is könnyedén személyes linuxos géppé alakíthatunk valamelyik akár névjegy CD-n vagy USB kulcson is elférő terjesztéssel.

Úgy őszintén, ***François***, a ***végső megoldás***, a mindenkori legjobb miért jelent mindig valami nagyobbat, gyorsabban és erőforrás-igényesebbet? ***Mon Dieu***, hiába a folyamatos gyorsulás, a végére valahogy mindig lassabban működnek a dolgok. Tudom, az általad ide az étterembe javasolt szuperszámítógépes fürt jó eséllyel indulna a hónap linuxos erőművének címéért, de sajnos nincs elég helyünk. Hogy? A pince? ***Non***, François, a borospincében bort tárolunk, és szeretném, ha ez így is maradna. Alighanem a vendégek is velem értenének egyet. Ha már szóba kerültek: bármelyik pillanatban itt lehetnek.
Ah, ***François***, már meg is érkeztek! Üdvözöl mindenkit a ***Chez Marcel***, a világ legfinomabb borainak, és természetesen a legjobb linuxos fogások otthona! Az asztalok készen állnak! Foglaljatok helyet, helyezzétek magatok kényelembe. A borospincébe, ***François***! Hozd fel, kérlek, a 2003-as német ***Auslese*** rizlinget! ***Vite!***
Míg készséges felszolgálóm kitölti a bort, gondolkozzunk el azon, mitől lesz egy linuxos gép a legjobb. ***François*** egy szuperszámítógépet javasolna. Nekem inkább valami kisebb, de semmivel sem haszontalanabb tetszene, valami olyan, ami akár a zsebemben is elfér. Nem egy alkalommal mentett már meg, hogy volt nálam egy ***Linux*** példány. Illetve, hogy pontos legyek, a megmentett személy egy valamilyen másfajta operációs rendszert futtató felhasználó volt, akinek olyan gondja akadt, amin csak linuxos rendszerrel lehetett segíteni. A magammal hordott miniterjesztések általában egy-két-három lemezen elfértek, és alapszintű, szöveges felületű segédeszközöket tartalmaztak. Ma két nagyszerű megoldást szeretnék mutatni arra, hogyan vihetjük magunkkal a ***Linuxot***, bárhová is induljunk. Ezek a miniterjesztések többé már nem a végletekig lecsupaszított, szöveges felületű eszközök halmazai, hanem teljesen grafikus, teljes értékű hálózatkezelésre képes terjesztések – mégis elférnek a zsebünkben vagy akár a tárcánkban is. A legjobb bennük az, hogy ***CD***-lemezről vagy ***USB***-kulcsról is futtathatjuk őket.
Mai menünk első fogása személyes kedvenceim egyike, a ***Damn Small Linux (DSL, „átkozottul kisméretű” Linux)***. A ***DSL*** egy ***Debian*** alapú, ***CD***-lemezről futtatható terjesztés. Mérete nem éri el még az 50 MB-ot sem, így akár egy hitelkártya méretű ***CD***-lemezen (ilyet szinte bármelyik számítógép- vagy irodaszerüzletben kapni) is elhelyezhető. Töltsük le az ***ISO*** képfájlt (lásd az internetes forrásokat), írjuk fel ***CD***-lemezre (ami szabványos vagy hitelkártya méretű is lehet), majd indítsuk újra a gépünket.
A ***DSL*** rendkívül gyors, kisméretű. Ablakkezelője a ***Fluxbox***. Szerény képességű gépeken is futtatható, akár 16 MB memóriával is megelégszik. Számos asztali alkalmazást tartalmaz, ezeket szintén a kevés erőforrás igénybe vételével történő, gyors futásra tervezték. Tartalmazza a ***Dillo*** és a ***Firefox*** webböngészőt, a ***Sylpheed*** levelezőprogramot, a ***Naim IRC***- és azonnali üzenetküldő ügyfelet, az ***XMMS*** zenelejátszót, grafikai szerkesztésre és ernyőképek készítésére



1. ábra A DSL felhasználói felülete sokoldalú, erőforrások tekintetében mégis igénytelen

az *Xpaintet*, szövegszerkesztésre az *FLwritert*, táblázatkezelésre a *Siaget* – egyéb feladatokra pedig számtalan más programot. A *DSL* működéséből az *1. ábra* kínál ízelítőt.

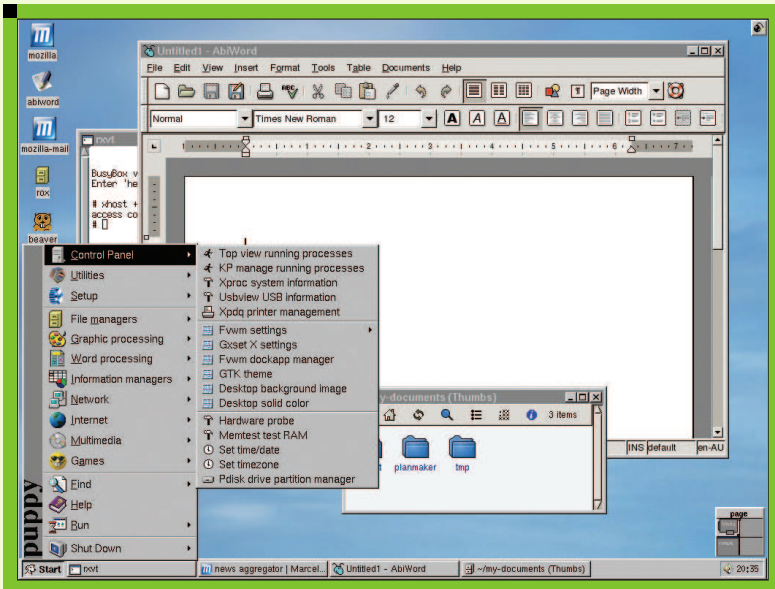
Ennél a terjesztésnél nincs a programok elérésére szolgáló gomb a bal alsó sarokban. Ha meg akarjuk jeleníteni a menüt, kattintsunk az asztal tetszőleges pontján az egér jobb gombjával. Ekkor megjelenik az alkalmazásmenü legfelső szintje, amely további almenükbe rendezve biztosítja a *DSL* összes szolgáltatásának elérését. Ha el akarjuk tüntetni a menüt, kattintsunk az egér bal gombjával az asztal egy üres pontjára.

Az egyik első teendőnk valószínűleg a hálózatkezelés beállítása lesz. Az egér jobb gombjával kattintva jelenítsük meg a menüt, melyből válasszuk a *System (Rendszer)*, majd a *Net Setup (Hálózat beállítása)* pontot. Módunk nyílik a telefonos hálózat, a hálózati kártya, a *DSL* (mármint az internetezős *DSL*) és korlátozott mértékben a vezeték nélküli hálózatok beállítására. Az *ndiswrapper* is a terjesztés része, hátha egy csak a *Microsoft* illesztőprogramjait támogató hálózati kártyába ütköznénk. A hálózati beállítások mindegyike menüből érhető el; egyszerűen csak adjuk meg a szükséges értékeket.

Ha már a rendszermenüben járunk, érdemes a *Daemons (Démonok)* részre is átváltani, ahol a *DSL* egy másik figyelemre méltó képességére ismerhetünk rá: *SSH* kiszolgálót, *NFS*-t, webkiszolgálót és *FTP*-kiszolgálót is találunk benne. A hagyományos *LPD* révén a nyomtatódémon-támogatás is megoldott.

Kis mérete ellenére a *DSL* még némi „díszítést” is tartalmaz. A *Desktop (Asztal)* menüből válasszuk a *Styles (Stílusok)* pontot, ahol többféle stílus közül választhatjuk ki, hogyan nézzen ki a munkafelület.

Mielőtt mai menünk következő fogására térnék, hadd hívjam fel a figyelmet az *Apps (Alkalmazások)* menü *Tools (Eszközök)* elemére. Tekintsünk alulra, ahol találunk egy a *DSL* merevlemezre végzett telepítését elindító menüpontot, illetve *USB*-kulcsra is telepíthetjük a terjesztést, így bárhova magunkkal vihetjük. Az *apt* és a *Synaptic* engedélyezésére is módunk nyílik, ezt követően további csomagokat is telepíthetünk. Ennek haszna nyilvánvaló, ha lemezre telepítjük a rendszert, de a *Tools* menü tetejére pillantva egy további indokot is találunk.



■ *2. ábra* Mindig is egy Puppy Linuxra vágytunk, nem igaz?

Ez pedig a *Make myDSL CD remaster (Saját DSL-példány készítése)* menüpont, mellyel saját, egyéni *DSL* változatot állíthatunk össze. Ha rákattintunk, egy újabb ablak jelenik meg, ez ismerteti, hogyan válthatunk át 2-es futási szintre, ahonnan elvégezhetjük a sokszorosítást. Gyakorlatilag annyit kell tennünk, hogy újraindítjuk a gépet, majd az indítási parancssorba begépeljük a `dsl toram 2` parancsot. Miután a héj parancssora megjelent, adjuk ki a `mkmydsl` parancsot. A folyamat ismertetéséhez nem áll rendelkezésemre elegendő hely, de a *www.damnsmalllinux.org/talk/node/113* oldalon minden szükséges tudnivaló megtalálható.

Szintén kisméretű, megismerésre érdemes, grafikus *Linux* terjesztés a *Puppy Linux*. Teljes értékű hálózatkezelésre képes, és szintén szép kis alkalmazásgyűjtemény tartozik hozzá. A hálózatkezelést illetően a *Puppy* webböngészésre *Mozillát*, levelezésre *Sylpheedet*, távfelügyeletre *SSH*-t, *VoIP* hívások lebonyolításához Gphone-t, a távoli asztalok megjelenítéséhez *VNC*-t és *rdesktop* ügyfelet tartalmaz – és még sok egyebet. Szövegszerkesztési célokra az *AbiWord* áll rendelkezésünkre, kiadványszerkesztésre pedig a *Scribust* vehetjük igénybe. A csomagot fájlkezelők, kép- és *HTML*-szerkesztők, táblázatkezelő és személyi pénzügyi program teszi teljessé.

Természetesen a játékok sem maradhattak ki belőle. Ha nagyon nem tudunk mit kezdeni magunkkal, a *Bubbles*, a *Frozen Bubble* leszármazottja és a *gtkfish* segít elütni az időt. Utóbbi egy fura kis játék, melyben selyempapírból készült hálóval indulunk halászni. Ha a kiszemelt hal túlságosan gyorsan úszik, amikor elkapjuk, akkor kiszakítja a hálót. A hálót a bal egérgombbal dobhatjuk a vízbe, majd próbáljunk lassabban úszó halakat elkapni – ehhez csak fel kell engednünk az egérgombot. Különös játék, és különösen jól el lehet időzni vele.

A *Puppy Linuxot* a webhelyéről tölthetjük le *ISO* képfájl formájában (lásd a forrásokat). Kedvenc *CD*-író programunkkal (esetemben ez a *K3b*) írjuk fel *CD* lemezre. A frissen írt lemezt tegyük be a meghajtóba, majd indítsuk újra a rendszert.

A *Puppy Linux* indításakor elsőként a billentyűzetválasztó képernyő jelenik meg. Én legördítettem az *us qwerty* elemre, majd megnyomtam az *Entert*. Ez után meg kell adnunk egerünk típusát. Nagy valószínűséggel nyugodtan meghagyhatjuk az alapbe-

3. ábra A bárhová magunkkal cipelhető Linux: Puppy Linux USB-kulcson

állítást, esetemben ez a *ps/2* volt.
A program arra is rákérdez, hogy egerünk rendelkezik-e a görgővel. Ha ezzel végeztünk, máris elindul a grafikus felület, amelyet választásunk szerinti módban, 640x480-as, 800x600-as és egyéb felbontásokban használhatunk. A felbontás azonnal változik, az *OK* gombra bármikor rákattintva elfogadhatjuk az aktuális értéket.
Ennyi. *Puppy Linux* rendszerünket sikeresen üzembe helyeztük. (*2. ábra*) Ezen a ponton akár a *CD*-lemezt is kivehetjük a gépből.
A *Puppy Linux* weboldalán azt állítják, hogy a terjesztés bármire telepíthető, merevlemezről, *ZIP* lemezről, hálózatról (vékony ügyfélként) és *USB*-kulcsról egyaránt képes futni, akár csak a *DSL*. Engem főként ez a lehetőség ragadott meg. Milyen érdekes ötlet: teljes értékű, grafikus *Linux* rendszer a zsebemben. Alap-összeállításában a *Puppy* túlságosan nagy ahhoz, hogy elférjen egy 50 MB-os névjegykártyán, ehhez némi igazításra van szükség (erről még lesz szó).
Kattintsunk a *Start* gombra, majd nyissuk meg a *Setup (Telepítés)* menüt. Ebben találunk néhány figyelemre érdemes lehetőséget, ezek egyike a *Puppy USB*-kulcsra való telepítése. Ha ezt választjuk, megjelenik egy párbeszédpanel, mely végigvezet bennünket a szükséges lépéseken, kezdve az *USB*-kulcs csatlakoztatásától, a megfelelő meghajtó (ha esetleg egynél több eszközünk is lenne) és a lemezrész kiválasztásán keresztül egészen a fájlok másolásáig. A másolás a merevlemezen lévő helyi fájlok alapján és a CD-lemezen lévő, a rendszerindításra használt *Puppy* alapján is elvégezhető.
A következő lépés végrehajtása eltart néhány percig, ekkor történik meg a különféle fájlok másolása (*vmlinuz*, *image.gz* és *usr_cram.fs*). A másolás befejezése után megadhatjuk a billentyűzet alapértelmezett nyelvét. Én az amerikait választottam, majd megnyomtam az *Entert*. Ezt követően még egy választást kell megtennünk, ez pedig a *Puppy* fájlrendszer tárolási módjának kiválasztása. Az első lehetőség *vfat* lemezrész használata, */rootként* befűzve, egyéb módosítások nélkül. A második lehetőség egy kisméretű *ext2* fájlrendszer létrehozása a lemezrészen. A második lehetőség használata az ajánlott, és ez hatékonyabb is, az elsőnek ugyanakkor megvan az az előnye, hogy fájljait *Windows* alól is el lehet érni. Én a másodikat választottam, majd megnyomtam az *Entert*.
Ezzel a *Puppy* telepítve van az *USB*-kulcsra, a rendszerindító parancsfájlt átírva jelszót is megadhatunk titkosított fájlrendszerhez. Kiváló ötlet, hiszen az *USB*-kulcsot könnyedén elveszíthetjük, esetleg ellophatják, és ilyenkor nem árt, ha el van látva némi védelemmel. Végül a telepítő indításra alkalmassá teszi az *USB*-kulcsot; ez után már indulhatunk is utunkra, zsebünkben a *Puppyval* (*3. ábra*).
Felhívnám rá a figyelmet, hogy nem minden számítógép képes *USB*-kulcsról rendszert indítani, bár a *BIOS* rendszertöltési beállításainak módosításával nyerünk némi esélyt a sikerre. Ha a számítógép nem támogatja az *USB*-s meghajtóról végzett rendszerindítást, akkor sem veszett el minden – feltéve, hogy van a gépben hajlékonylemezes meghajtó. A *Puppy* webhelyén található egy rendszerindító képfájl (*boot2pup.img.gz* névvel), ezt másoljuk át hajlékonylemezre. Bontsuk ki, majd végezzük el a másolást:

```
gunzip boot2pup.img.gz
dd if=boot2pup.img of=/dev/fd0
```

Most már csak arra kell ügyelnünk, nehogy otthon felejtsük a lemezt. Mielőtt elbúcsúznánk a *Puppy Linuxtól*, szeretném megemlíteni egy másik kiváló szolgáltatását. A *Setup* menüben található egy *Remaster Puppy live-CD (A Puppy CD sokszorosítása)* pont is. Ez egy egyszerű parancsfájl, mely végigvezet minket a meglévő *CD*-lemez memóriába másolásához szükséges lépéseken (ehhez legalább 256 MB memóriára van szükség), a fájlrendszer szerkesztésén, a képfájl előállításán és *CD*-lemezre írásán.
Az első próbálkozásunk talán nem lesz sikeres, de ne adjuk fel, nem rossz program ez. Az egyik furcsa lépés a *CD*-író és az olvasó kiválasztásának megerősítése. Ezen a ponton a *Puppy* újraindul (egy *CD*-lemezről futó terjesztés esetében ez meglehetősen szokatlan), és bekapcsolja a *SCSI* emulációt. Miután a rendszer újraindult, lépjünk vissza a *Setup* menübe, és indítsuk újra a sokszorosító programot. Azonnal a harmadik lépésre kell ugrania, ahol be kell tennünk a *CD* lemezt az olvasóként kiválasztott meghajtóba. Ezt egy kérdés-válasz menet követi, melyben pontosan meg kell adnunk, hogy az elkészítendő *Puppy* változat hogyan nézzen ki.
Már említettem, hogy a programmal barátkozni kell egy kicsit, de próbáljuk könnyedén venni a dolgot, így észre sem fogjuk venni, és rövid idő alatt profin fogjuk kezelni. Ha megvan az új *ISO* képfájl, a *Puppy* elindítja a *Gcombust CD*-író programot, és befejezi a munkát.
Mon Dieu! Már ennyi idő van? Ha hihetünk az órának, bizony elérkezett a zárás pillanata. Csak semmi kapkodás! Pihenjetek még egy kicsit, biztos vagyok abban, hogy *François* örömmel tölti újra poharatokat. Fogjatok egyet a névjegykártya méretű lemezek közül, és főzzetek ki egy kis *Linuxot*, amit majd magatokkal vihettek. Emeljük poharunkat, *mes amis*, és igyunk egymás egészségére!
A *votre santé! Bon appétit!*

Linux Journal 2005. 136. szám

A cikkhez tartozó források elérhetősége:
➲ *www.linuxjournal.com/article/8326*

Marcel Gagné
(mggagne@salmar.com) Mississaguában, Ontario államban él. Ő a szerzője a Kiskapu kiadásában tavaly szeptemberben megjelent Linux-rendszerfelügyelet (ISBN 96-9301-40) című könyvnek.

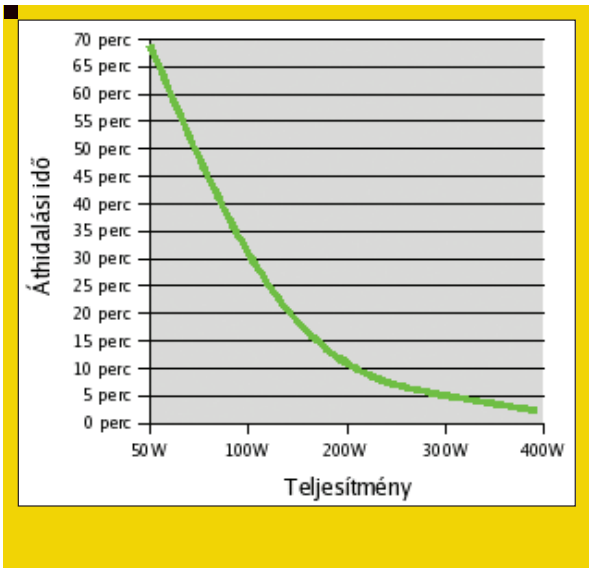
FreeBSD – a szomszéd vár (11. rész)

A szünetmentes tápellátás

A számítógépek megfelelő tápellátást igényelnek, amelyet a helyi áramszolgáltatótól tudunk „beszerezni”. Ezen tápellátás – akár pillanatnyi – kihagyása bosszúságot és elveszett munkát jelent; különféle zavarai pedig megmagyarázhatatlan problémákat okozhatnak, akár a hardver meghibásodását is (gondolva itt a másodlagos villámkárokra).

UPS típusok

Semmi sem olyan egyszerű, mint aminek látszik, így az ***UPS*** (***Uninterruptable Power Supply)*** eszközöknek is több típusa létezik, amelyek más-más célra szolgálnak. Alapvetően öt tulajdonságát említhetjük meg ezen eszközöknek: a kimeneti feszültség jellegét, a terhelhetőségét, az áthidalási időt, a működés módját, illetve géppel való kommunikáció jellegét. A minőség mellett ezen jellemzők befolyásolják a szünetmentes tápegység árát, s akár nagyságrendnyi különbséget is jelenhetnek. Régebben érdemes volt a túlfeszültség és villámvédelemre is figyelni, azonban a mostanában kapható készülékek ezt már szinte mind tartalmazzák.
A ***kimeneti feszültség*** az olcsóbb készülékek esetén egyszerű négyszög, amely azonban ideális a számítógépeink számára, s egyéb eszközök esetén – az utóbbi időben előszeretettel alkalmazott kapcsolóüzemű tápegységek is elviselik. Ez az olcsóság tehát nem jelent különösebb hátrányt, a legtöbb UPS ilyen kimenettel rendelkezik. Néhány eszköz (aktív hálózati eszköz, nyomtató, stb.) azonban szinusz (vagy kvázi-szinusz) feszültséget vár, s ha négyszöget kapnak, képesek jelentősen melegedni és ezáltal idővel meghibásodni, illetve bizonytalanul működni.
A szünetmentes ***terhelhetőségét*** úgy kell megválasztanunk, hogy a számítógépünk, illetve a hozzá tartozó egyéb perifériák összteljesítményét elérje. Nehéz meghatározni ezt az összteljesítményt, mivel sok feltételtől függ, a legbiztosabb módszer, ha a számítógépünk tápegységének teljesítményéhez hozzáadjuk a monitor teljesítményét, s ezt tekintjük irányszámnak a vásárlás során. Ha pénztálcánk engedi, akkor vásárolhatunk nagyobb terhelhetőségű ***UPS***-t is, mint amire pillanatnyilag szükségünk van. Egy átlagos munkaállomáshoz elég 350VA/210W, egy combosabb munkaállomáshoz azonban már elkel minimum 600VA/360W terhelhetőség.
A maximális ***áthidalási idő*** általában nem választható szabadon, az egyszerűbb szünetmentes tápegységek esetén többnyire összefügg a terhelhetőséggel: minél jobban terhelhető egy ***UPS***, annál tovább képes helytállni a névlegesnél kisebb teljesítményen, az áthidalási idő ugyanis fordítottan arányos az aktuális terheléssel. Ez a gyakorlatban azzal jár, hogy teljes terhelésen mindössze 2-5perc áll rendelkezésünkre, mielőtt gépünk végleg tápellátás nélkül marad; ellenben 10%-15% körüli terheléssel akár egy órán át is kitarthat! Nevesebb gyártók mellékelnek táblázatot vagy grafikont, amelyről leolvashatjuk ezeket az összefüggéseket, és a számunkra megfelelő ***UPS***-t vásárolhatjuk meg.
A szünetmentes ***működésének módja*** alapvetően kétféle lehet: off-line és on-line. Az elterjedt (és olcsóbb) megoldás az első variáció, amely esetén az eszközeink úgy kapják a tápellátást, mintha nem lenne szünetmentes tápegységünk, és az ***UPS*** csak a hálózati feszültség nagymértékű rendellenessége esetén kapcsol át akkumulátoros üzemmódra. A legtöbb hálózati zavart nem szűri ki,



viszont védelmet nyújt a túlfeszültség és a feszültségkimaradás ellen, s ez többnyire elég is. A másik módszer több altípusra osztható, ezek kismértékben képesek a bemeneti feszültséget a kívánalmaknak megfelelően átalakítani (***line-interactive***), illetve akár állandóan akkumulátoros üzemben dolgozva minden zavart kiszűrnek (kettős konverziójú); s természetesen ezt a minőséget meg is kell fizetni.

Lényeges tulajdonság a ***számítógéppel való kommunikáció*** is, amely nagy mértékben meghatározza a szünetmentes tápegység használati körét. Ha nem számítógépet (elektromos írógép, pénztárgép, stb.) kell megvédeni, akkor használhatunk egyszerű szünetmentes tápegységeket, amelyek általában semmilyen kommunikációra nem képesek, egyszerűen a berendezés előtt ülő felhasználóra bízzák annak megítélését, hogy mikor állítsa le a gépét. Számítógépek esetén célszerű olyan ***UPS***-t választani, amelyek képesek egyszerűbb kommunikációra, például egyszerűen csak azt közlik a számítógéppel, hogy van-e hálózati feszültség, vagy nincs: a kimaradás időtartamának számontartását és az időzítések kezelését az operációs rendszerre bízzák. A legfejlettebb megoldás során két irányban is folyik kommunikáció, az ***UPS*** közölheti a töltöttségi paramétereket, a pillanatnyi terhelést, illetve akár a bemenő hálózati feszültség paramétereit is. A kétirányú kommunikáció előnye, hogy az operációs rendszer képes vezérelni a szünetmentes tápegység kimeneteit is: ha éppen nincs nyomtatnivaló, akkor a nyomtatót felesleges akkumulátorról táplálni. A régebbi ***UPS*** soros kapcsolaton át végezte ezt az adatcserét, viszont manapság már szinte kizárólag csak ***USB*** csatlakozóval kapni szünetmentes tápegységet.

Néhány egyéb kényelmi szolgáltatást is észrevehetünk bizonyos ***UPS***-ek esetén, a többség rendelkezik olyan kimenettel is, amely csak túlfeszültség védett, és így megoldható, hogy áramkimaradás esetén soha nem biztosítunk szünetmentes tápellátást bizonyos készülékeknek (fax, nyomtató, stb.); továbbá szinte kivétel nélkül találhatunk két telefoncsatlakozót (egy a fali csatlakozónak és egy a telefon/fax/modem felé), amelyekkel a telefonvonalunkat védhetjük meg a túlfeszültségtől.

FreeBSD és Linux

Általában kevés dologra kell figyelni, ha az elterjedt ***Windows*** operációs rendszerhez szeretnénk szünetmentes tápellátást, mivel az összes készülékhez kapunk szoftvereket, amelyekkel ez lehetővé válik. ***Linux*** rendszerhez már kevesebb ***UPS*** illeszthető, de egyre több telepítő CD-n találni ***Linux*** alá írt kezelőszoftvereket, amelyeket az elterjedt disztribúciók is már tartalmaznak. Sajnos ***FreeBSD*** rendszerhez nagyon kevés szünetmentes táp kapcsolható.

Ha biztosra akarunk menni, akkor az ***APC*** cég termékei közül válasszunk egy számunkra megfelelő készüléket, mivel ehhez létezik olyan program, amellyel teljeskörűen tudjuk használni az ***UPS***-t: ez az ***APCUPSD***. Választhatunk más gyártó termékei között is, sok „***Noname***” gyártó másolja az ***APC*** kommunikációját, illetve létezik néhány egyéb gyártó által nevesített ***APC*** termék is (például néhány ***HP UPS APC*** gyártmány). Sok gyártó saját szoftvert és saját kommunikációs protokollt készít a szünetmentes tápegységéhez, s ha nem ad hozzá specifikációt, esetleg ***Linux*** vagy ***FreeBSD*** programot, akkor nem tudjuk érdemben használni.

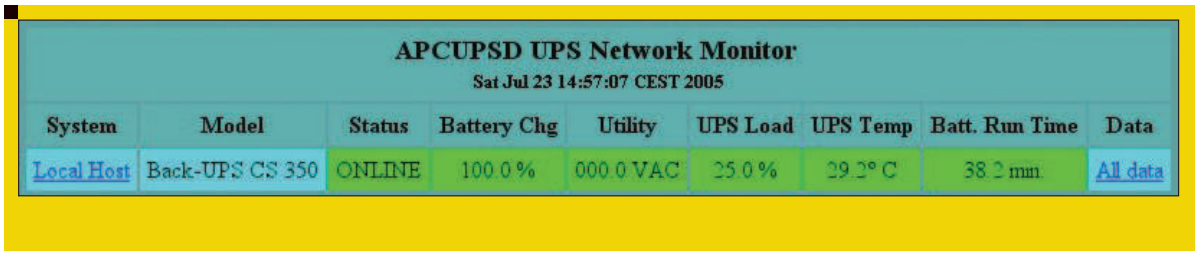
Az APC UPS Daemon

Ha ***APC*** szünetmentes tápegységet választottunk, akkor itt az idő, hogy feltelepítsük a hozzá tartozó kezelőprogramot, amelyet többnyire `apcupsd` néven találunk meg. ***FreeBSD*** esetén a ***sysutils/apcupsd*** helyen találjuk meg, s a szokott módon tudjuk telepíteni. A program teljeskörűen képes kezelni bármelyik ***APC UPS*** készüléket, amelyiket össze tudjuk kötni a gépünkkel.

A legegyszerűbb kommunikáció a ***dumb*** kapcsolat, amely egyszerűen csak azt jelzi a gép felé, hogy van-e hálózati feszültség. Ezek a típusok lassan kihalnak, újonnan már nem tudunk ilyen szünetmentes tápegységet beszerezni, hiszen csak a soros (***RS232***) kapcsolattal rendelkező ***UPS***-ek között voltak ilyenek. Ha az ***UPS USB*** kábellel kapcsolódik a gépünkhöz, akkor ne állítsunk be ilyen típust, nem fog működni. Jól nézzük meg a kereskedőnél a kapható típusokat, mert ilyen ***UPS***-t már nem ildomos vásárolni, csak nehézségeink lesznek vele.

A ***dumb*** típust követte az ***apcsmart*** nevű kapcsolat, amely még mindig soros kapcsolat volt kis sebességű (2400bps) kétirányú adatátvitellel (a számítógép oldaláról gyakorlatilag modemként lehetett kezelni ezeket a szünetmentes tápegységeket). Eleinte kevés parancsot ismertek az eszközök, idővel ezek finomodtak és bővültek. Az ***USB*** megjelenésével ez a típus is lassan feledésbe merül, s hasonlóan a ***dumb*** típushoz: ne vegyünk soros kapcsolattal szünetmentes tápegységet, hacsak nincs valami nyomós okunk rá (az operációs rendszer vagy a védendő számítógép nem támogatja). Sok egyéb gyártó használ az ***APC***-vel azonos parancsokat, így nagy esélyünk van arra, hogy az ***APCUPSD*** kezelni képes más ***UPS***-t is.

A legújabb kommunikáció az ***usb*** kapcsolattípus, amely nagyjából két éve jelent meg a drágább modellek között, s mostanára az összes ***UPS*** ilyen kommunikációval érkezik.



APCUPSD UPS Network Monitor

Sat Jul 23 14:57:07 CEST 2005

System	Model	Status	Battery Chg	Utility	UPS Load	UPS Temp	Batt. Run Time	Data
Local Host	Back-UPS CS 350	ONLINE	100.0 %	000.0 VAC	25.0 %	29.2° C	38.2 mn.	All data

A többi gyártó is ***USB*** felülettel készíti már a készülékeit, sajnos közös szabvány hiányában a használt protokollban jelentős eltérések is lehetnek, így ***USB*** felületen szinte csak *APC* gyártmányú ***UPS***-t leszünk képesek használni.
A közvetlen kapcsolatokon túl létezik még kapcsolati módszer, amely esetén hálózaton át tudunk kommunikálni a szünetmentes tápegységgel. Léteznek olyan ***UPS***-ek, amelyek fel vannak készítve ***Ethernet*** alapú kommunikációra, akár egy kisebb webszerver is fut bennük, amelyen át le tudjuk kérdezni az állapotukat, de vezérelni is tudjuk szükség esetén. A webszerver mellett többnyire van ***SNMP (Simple Network Management Protocol)*** felület, amely már programok számára is használható.
Ha több gépet kötünk egy ***UPS***-re, akkor vagy az ***snmp*** típust kell kiválasztanunk, ha erre nem képes a szünetmentes tápegység, akkor az ***APCUPSD*** képes ezt a problémát is áthidalni. Minden ***UPS*** által védett gépre fel kell telepítenünk az ***APCUPSD*** programot, csak más módon kell beállítanunk minden egyes példányt. Kinevezzük ***Master*** státuszra azt a gépet, amelyik közvetlenül csatlakozik a szünetmentes tápegységre, s a többi gépünk ***Slave*** lesz. A „mester” megosztja az információkat a „szolgákkal”, így áramkimaradás esetén a „szolga” gépek ugyan úgy értesülnek mindenről, mintha mindegyik önálló szünetmentessel lenne kapcsolatban. Ez utóbbi esetben a ***net*** lesz a kapcsolódás típusa.
Látható, hogy az ***APCUPSD*** három állapotot ismer: ***standalone***, amikor egyetlen gép csatlakozik a szünetmenteshez; ***master***, ha megoszt információkat más gépekkel; ***slave***, ha rá van kötve egy ***UPS***-re kötve, de nincs vele közvetlen kapcsolata.

Standalone

Miután összekötöttük a szünetmentes tápot a gépünkkel, első dolgunk, hogy megkeressük annak lenyomatát a rendszerüzenetek között. Soros kapcsolat esetén csak annyit kell tudnunk, hogy melyik portra kötöttük, s azt a portot hogy hívják a használt rendszer esetén (például */dev/ttyS0* vagy */dev/cuaa0*). ***USB*** kábel kihúzása és újra bedugása már a naplóbejegyzések között is nyomot hagy, például a `dmesg` parancs hatására látnunk kell néhány hasonló sort:

```
uhid0: American Power Conversion Back-UPS CS 350
➥FW:807.q5.I USB FW:q5, rev 1.10/0.06,
addr 2, iclass 3/0
uhid0: at uhub0 port 1 (addr 2) disconnected
uhid0: detached
uhid0: American Power Conversion Back-UPS CS 350
➥FW:807.q5.I USB FW:q5, rev 1.10/0.06,
addr 2, iclass 3/0
```

Linux esetén nincs különösebb gond, viszont a ****BSD*** rendszerek nem kellő körültekintéssel kezelik ezt az ***USB*** kapcsolatot, s az `uhid` meghajtó ráül a kommunikációs vonalra, amely így alkalmatlan arra, hogy az ***APCUPSD*** kezelni tudja. ***FreeBSD*** esetén ezt a meghajtót el kell távolítanunk a rendszermagból, egyszerűen a konfigurációs állományban (például ***GENERIC***) komment jelet kell tennünk a

```
device uhid
```

sor elé, majd új rendszermagot kell készítenünk és már újra is indíthatjuk a gépet. Dolgunkat akkor végeztük jól, ha a

```
ugen0: American Power Conversion Back-UPS CS 350
➥FW:807.q5.I USB FW:q5, rev 1.10/0.06,
addr 2, iclass 3/0
```

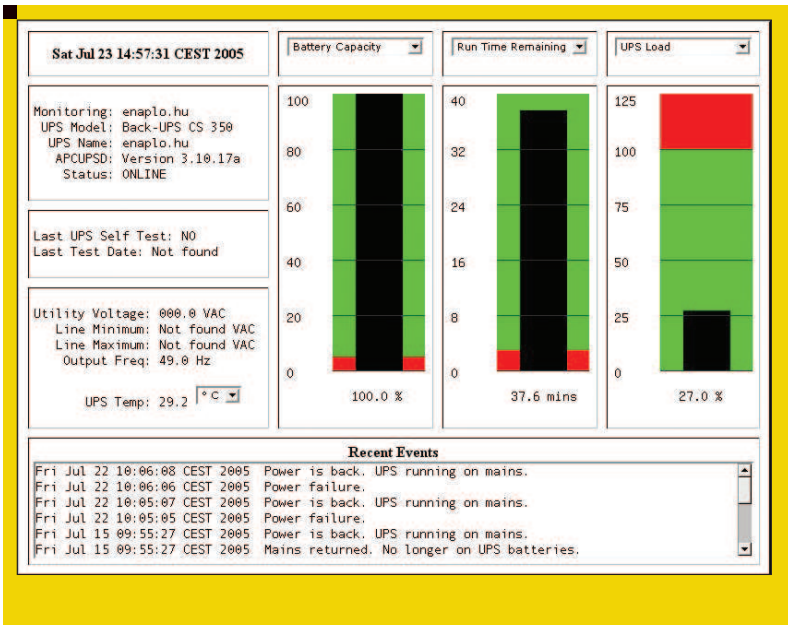
sort látjuk, vagyis uhid0 helyett már ugen0 eszközt! Mivel úgyis újraindítjuk a gépet, közben ellenőrizhetjük, hogy bekapcsol-e automatikusan ha hálózati feszültséget kap a tápja. A jelenleg kapható ***Intel*** alapú ***PC*** jellegű számítógépek általában nem kapcsolnak be ilyen esetben újra, viszont a ***BIOS SETUP*** programjában beállítható ez a viselkedés (általában „power” és „restart” szöveget kell keresnünk). Az ***UPS*** csak akkor tudja ***visszakapcsolni*** a számítógépet, ha az képes a hálózati feszültség megérkezésekor saját maga elindulni.
Az ***APCUPSD*** egy minta konfigurációs állománnyal rendelkezik, amelyből a sok felesleges sort kigyomlálva egy egyszerű állományt tudunk készíteni, mint például:

```
UPSCABLE usb
UPSTYPE usb
UPSCLASS standalone
UPSMODE disable
DEVICE
LOCKFILE /var/spool/lock
ONBATTERYDELAY 6
BATTERYLEVEL 5
MINUTES 3
```

A démont elindítva már láthatjuk is a naplóbejegyzések között az új sort

```
apcupsd[1095]: apcupsd 3.10.17a (22 April 2005)
➥freebsd startup succeeded
```

amely arról tájékoztat minket, hogy az `apcupsd` megfelelően elindult. A hibás konfigurációs állomány viszont rejthet néhány meglepetést is:




```
MAXTIME  : 0 Seconds
OUTPUTV  : 230.0 Volts
DWAKE    : 000 Seconds
DSHUTD   : 000 Seconds
LOTRANS  : 29364.0 Volts
HITRANS  : 266.0 Volts
ITEMP    : 29.2 C Internal
ALARMDEL : Always
BATTV    : 13.6 Volts
LINEFREQ : 49.0 Hz
NUMXFERS : 4
XONBATT  : Fri Jul 22 10:06:06
           ➥CEST 2005
TONBATT  : 0 seconds
CUMONBATT: 12 seconds
XOFFBATT : Fri Jul 22 10:06:08
           ➥CEST 2005
SELFTEST : NO
STATFLAG : 0x02000008 Status
           ➥Flag
SERIALNO : BB0503005874
BATTDATE : 2005-01-11
NOMBATTV :  12.0
FIRMWARE : 07.q5.I USB FW:q5
APCMODEL : Back-UPS CS 350
END APC  : Sat Jul 23 14:19:56 CEST 2005
```

```
apcupsd[9512]: apcupsd FATAL ERROR in bsd-usb.c
➥at line 781 Cannot find UPS device
apcupsd[9524]: apcupsd FATAL ERROR in apcipc.c
➥at line 118 Mutex lock failure. ERR=Resource
deadlock avoided
kernel: pid 9524 (apcupsd), uid 0: exited on
➥signal 10 (core dumped)
apcupsd[9573]: apcupsd FATAL ERROR in
➥smartsetup.c at line 177 PANIC! Cannot
➥communicate with UPS via serial port. Please
➥make sure the port specified on the DEVICE
➥directive is correct, and that your cable
➥specification on the UPSCABLE directive is
➥correct.
```

Ha sikeresen beüzemeltük a démont, akkor lekérdezhetjük a szünetmentes tápegységünk állapotot is, mégpedig az `apcaccess status` parancs segítségével:

```
APC      : 001,039,1004
DATE     : Sat Jul 23 14:19:52 CEST 2005
HOSTNAME : enaplo.hu
RELEASE  : 3.10.17a
VERSION  : 3.10.17a (22 April 2005) freebsd
UPSNAME  : enaplo.hu
CABLE    : USB Cable
MODEL    : Back-UPS CS 350
UPSMODE  : Stand Alone
STARTTIME: Mon Jul 11 20:39:32 CEST 2005
STATUS   : ONLINE
LINEV    : 000.0 Volts
LOADPCT  :  26.0 Percent Load Capacity
BCHARGE  : 100.0 Percent
TIMELEFT :  38.2 Minutes
MBATTCHG : 5 Percent
MINTIMEL : 3 Minutes
```

Minél drágább a szünetmentes tápegység, annál több információt szokott közölni a működési paramétereiről, de a legfontosabb három paramétert mindegyik képes kijelezni: az aktuális terhelést, az akkumulátorok töltöttségét és a megbecsült áthidalási időt (`LOADPCT`, `BCHARGE` és `TIMELEFT`).
Itt az ideje, hogy kipróbáljuk a működőképességét a szünetmentes tápunknak, lehetőleg akkor, ha kevesen dolgoznak a védett gépen. A teszt abból áll, hogy az ***UPS***-t kihúzzuk a fali konnektorból, majd kis idő múlva visszadugjuk oda (elég egy másodperc is akár). Ha az ***UPS*** és a gép között a kommunikáció megfelelő, akkor látnunk kell bejegyzéseket mind a gép naplójában, mind az ***APCUPSD*** saját naplójában (ha be van állítva)

```
apcupsd[1095]: Power failure.
apcupsd[1095]: Running on UPS batteries.
apcupsd[1095]: Mains returned. No longer on UPS
               ➥batteries.
apcupsd[1095]: Power is back. UPS running on
               ➥mains.
```

Ezeken túlmenően az összes nyitott konzolon üzeneteket kell látnunk:

```
Broadcast Message from root@enaplo.hu
        (no tty) at 10:05 CEST...
Warning power loss detected.

Broadcast Message from root@enaplo.hu
        (no tty) at 10:05 CEST...
Power has returned...
```

Ha a hálózati feszültség nem jön vissza, akkor a szünetmentes tápegység akkumulátorai kezdenek veszíteni a töltöttségükből, s ezzel arányosan csökken az áthidalási idő is. A konfigurációs állományban meg tudunk adni határértékeket, amelyeket elérve megkezdődik a számítógép leállítása. Alapesetben 5% akkumulátortöltöttség, illetve 3 perc áthidalási idő alatt kezdődik meg az operációs rendszer és a gép leállítása. Ha nagyobb adatbázisok is vannak a kiszolgáló gépen, akkor vegyük nagyobbra a legkisebb áthidalási időt, mert esetleg a programnak nem lesz elég ideje lezárni a megkezdett tranzakciókat és sérült vagy inkonzisztens állapotban maradhat az adatbázis.
Az ***APCUPSD*** tartalmaz három ***CGI*** programot is, amelyeket át tudunk másolni a webszerverünk megfelelő helyére, s így távolról is kényelmesen megnézhetjük a szünetmentes ellátás paramétereit.
A ***multimon.cgi*** dolga, hogy egy táblázatba foglalja a helyi hálózat összes elérhető szünetmentesének jellemzőit. Ahhoz, hogy az ***APCUPSD*** tudja, hogy mely gépeket kell megkérdeznie, a konfigurációs állományban fel kell sorolnunk ezen gépeket:

```
MONITOR szerver1 "Adatbázis szerver"
MONITOR szerver2 "Webszerver"
```

Ha csak egy ilyen található, akkor egy sort láthatunk. *(2. ábra)*

Ha rákattintunk a ***„System"*** oszlop egy-egy cellájára, akkor egy beszédesebb grafikont is láthatunk, amelyből kiolvashatjuk az adott szünetmentes állapotát, ahol különféle grafikonok között is tudunk váltogatni. *(3. ábra)*
Ha visszatérünk a táblázatos formához, akkor az ***„All Data"*** oszlop megfelelő cellájára kattintva az ***UPS*** összes adatát is le tudjuk kérdezni, amely azonos lesz az `apcaccess status` által visszaadott sorokkal.
Érdemes feldolgozni és letárolni a lekérdezhető adatokat, így később érdekes statisztikákat is el tudunk készíteni, havi, heti vagy akár napi fogyasztást is tudunk összesíteni (például ***http://www.enaplo.hu/index.jsp?page=visitor.upsMonth*** vagy ***http://www.enaplo.hu/index.jsp?page=visitor.upsDay***).

Master

Az ***APCUPSD*** „mester" működése nagyon gyorsan beállítható, egyszerűen az ***UPSCLASS*** és az ***UPSMODE*** sorokat kell megváltoztatni, majd leállítani és újra elindítani a démont.

```
UPSCABLE usb
UPSTYPE usb
UPSCLASS netmaster
UPSMODE net
DEVICE
LOCKFILE /var/spool/lock
ONBATTERYDELAY 6
BATTERYLEVEL 5
MINUTES 3
```

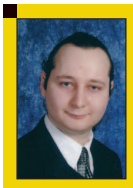
Slave

Az ***APCUPSD*** „szolgák" beállítása sem igényel túl sok időt, egyszerűen csak tudatosítani kell, hogy „szolga" lesz és meg kell adnunk a „mester" elérhetőségét. Figyeljünk arra, hogy az UPS kábel típusa már „ether", mivel nincs közvetlen kapcsolatban a szünetmentes tápegységgel, s az eszközt leíró sort se adjuk meg.

```
UPSCABLE ether
UPSTYPE smartups
UPSCLASS netslave
UPSMODE net
NETPORT 6666
MASTER 192.168.1.1
LOCKFILE /var/spool/lock
ONBATTERYDELAY 6
BATTERYLEVEL 5
MINUTES 3
```

Karbantartás

Fordítsunk gondot a szünetmentes táp karbantartására, ne várjuk meg azt a jelenséget, hogy egy 100% töltöttséget jelző akkumulátor másodperceken belül lemerül. Rendszeresen teszteljük az ***UPS*** működését, és nézzük figyelmesen a töltöttség csökkenését az áthidalási idő és terhelés függvényében. Ha indokolatlanul gyorsan merül az akkumulátor, akkor hamarosan elérkezik az ideje a cserének, ne várjuk meg, amíg az ***UPS*** jelzi ezt.
Soha ne terheljük 70-80% felett az ***UPS***-t! Bár a ráírt teljesítményig terhelhető, nem érdemes határértéken járatni, mind az elektronika, mind a akkumulátor élettartama csökken a terhelés növelésével. Bár elviseli a túlterhelést is a szünetmentes tápegység, csak indokolt esetben terheljük túl.
Ügyes kezekkel megoldható az áthidalási idő kitolása, ha nagyobb kapacitású akkumulátort teszünk a szünetmentes tápunkba, mint amit előírtak. A legtöbb ***UPS*** képes tölteni nagyobb akkumulátort is, mint amellyel gyárilag szerelik, viszont csak kevés képes felismerni a nagyobb kapacitást és azzal áthidalási időt számolni.



Auth Gábor (auth.gabor@enaplo.hu)
Egy pécsi középiskolában informatikát és programozást oktat. Tíz éve botlott először a UNIX rendszerekbe, 7 év Linux használat után kapta el a FreeBSD lázat, amiből máig nem tudott kigyógyulni.

KAPCSOLÓDÓ CÍMEK

A FreeBSD projekt honlapja: ➲ http://www.freebsd.org

A magyar FreeBSD honlap: ➲ http://www.freebsd.hu

A magyar BSD honlap: ➲ http://www.bsd.hu

A kézikönyv magyar fordítása

➲ http://www.enaplo.hu/FreeBSD/handbook/

Nagyban kicsit más – Linux az oktatási szférában

Legtöbb rendszergazdának a nyár a szabadságot, a nyaralást jelenti. Nem feltétlen van ez így az oktatási szférában. Ilyenkor kell ugyanis a következő félévre – vagy évre – felkészíteni mind az oktatói, mind a hallgatói használatra szánt számtógépeket...

A GRID-es rendszergazdáknak mindig könnyebb dolguk van, ők ugyanis általában sok ugyanolyan kiépítésű és szoftverigényű rendszert telepítenek. De mi a helyzet akkor, ha mondjuk egy egyetem informatikai tanszékcsoportja a kétszáz számítógépét nem egyszerre tudja/tudta beszerezni? Ekkor jó esély van arra, hogy több különböző „garnitúra" masina áll a rendelkezésünkre. Nálunk szerencsére a vezetőség figyelt arra, hogy lehetőleg az összes gép ***Linux***-kompatibilis elemekből álljon, ami persze nem jelenti azt, hogy egy-egy alkatrésszel esetenként ne szenvedtünk volna rendszergazda kollégáimmal vállvetve.

Előkészületek

Mielőtt hozzákezdenénk, érdemes összekészíteni pár dolgot:

- Csavarhúzó – Ha ki kell lesni egy-egy alkatrész típusát...
- Linux telepítőlemezek – A ***Fedora*** első négy telepítőlemeze, vagy ***Debianból*** a ***Netinst CD***
- g4u – ***Ghost4Unix*** bootfloppy vagy boot CD.
- memtest86 – Memóriatesztelő program, ha minden kötél szakad...
- Egy-két Knoppix Live CD – Még jól jöhet...
- Némi gyakorlat – Nagy adag türelemmel fűszerezve...

Tegyük fel, hogy a gép, amit kaptunk „teljesen szűz". A ***Knoppix*** indítása után partícionáljuk a merevlemezt, gondolva arra is, hogy esetleg más operációs rendszer is kerülhet majd rá. Nem érdemes a ***Linux*** rendszerünknek 15 gigabájtnál nagyobb tárterületet hagyni, a ***csereterület (swap partíció)*** méretét pedig úgy érdemes megválasztani, hogy a valódi ***RAM*** és a ***swap*** méretének összege ne legyen több másfél-két gigabájtnál. A ***boot*** partíciónak (amit majd a ***/boot*** könyvtár alá fogunk becsatolni) legfeljebb 100 megabájtot érdemes hagyni. A merevlemez partíciós táblája tehát valahogy így fog kinézni:

```
Device    Size      Mountpoint
hda1      100 Mb    /boot
hda2      ??? Gb    másik operációs rendszer
hda5      15 Gb     /
hda6      512 Mb    <swap>
```

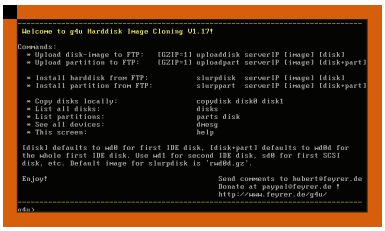
Véleményem szerint jelenleg két olyan disztribúció van, amelyek megfelelnek azoknak, aki hozzánk hasonlóan „nagyban játszanak": a ***Fedora Core*** és a ***Debian***. Ennek a legfőbb oka az, hogy ezeknél a legkisebb a valószínűsége annak, hogy forrásból kelljen fordítanunk valamelyik az oktatók és a hallgatók által igényelt szoftvert. Az egyéb szempontok alapján ugyanakkor jó választás lehet az ***Ubuntu Linux***, a ***Novell Desktop Linux (NLD)*** vagy az ***UHU-Linux*** is. Azt is lényeges szem előtt tartani, hogy ha egy disztribúciót könnyű karbantartani egyetlen gépen, abból nem következik, hogy ugyanilyen könnyű dolgunk lesz 100-200 gép esetén. Ilyen célra egyszerűen, vagyis héjprogramokkal manipulálható disztribúciót érdemes választani.

A sokgépes rendszerben hasznos elem egy megfelelően beállított ***DHCP***, ***LDAP*** és ***NFS*** kiszolgáló. A ***DHCP*** szerver előnye: ha esetleg megváltozik a DNS névkiszolgáló címe, akkor a megfelelő információt nem 200 gépen kell átírni, hanem csak a ***DHCP*** szerveren. Az ***LDAP*** kiszolgálóra szintén emiatt van szükség, hiszen így nem kell a sok gép mindegyikén regisztrálni minden felhasználót, nem is beszélve arról a szituációról, ha valaki az egyik gép előtt ülve éppen jelszót cserél. Ilyenkor az ***LDAP*** automatikusan elintéz mindent, így emberünk az összes gépre azonnal az új jelszóval tud majd belépni. Végül az ***NFS*** azért kell, mert így egyrészt nem kell a 200 gép között a felhasználók ***home*** könyvtárát folyamatosan szinkronizálni, másrészt a munkaállomásokba bőven elég egy 20-40 gigabájtos merevlemez még háromezer felhasználó esetén is.

Kezdődik a tényleges telepítés

Miután ezek megvannak, elkezdhetjük a telepítést. Ez nagyjából úgy zajlik, mintha a saját munkaállomásunkat

telepítenénk. Érdemes azonban többféle ablakkezelőt, többféle böngészőt, és többféle szerkesztőt feltenni. Az emberek ugyanis sokfélék, minden diák mást kedvel. Az se hátrány, ha felrakunk néhány *CD*- és médialejátszó programot is, elvégre sokan szeretnek munka közben zenét hallgatni, akár hálózaton át, akár hozott *CD*-ről.



Nálunk még fel szokott kerülni pár extra, ami a *Linux* telepítő *CD*-ken nincsenek fenn. Ilyenek például: *Adobe Reader*, *Mplayer*, *Macromedia FlashPlayer*, *Java Software Development Kit*, illetve pár fizetős program: *Maple*, *Matlab*. Ez utóbbiakhoz kell licenszszerver, ugyanis nem célszerű egy meghatározott gépcsoportra telepíteni. Sokkal jobb, ha a licenszszerver osztja ki a licenszeket, hogy a hallgató/oktató igény esetén bármelyik gépnél tudjon dolgozni vele, nyilván csak a maximális licenszszám erejéig.
Miután ezekkel megvagyunk, érdemes még a floppy és a *CD-ROM* használati jogosultságait megfelelően beállítani, illetve egy *csatolási pontot (mountpointot)* létrehozni a pendrive-nak. Az *NFS*-en megosztott *home* könyvtárat is itt érdemes beállítani. Tehát az */etc/fstab* fájl valahogy így fog kinézni:

```
/dev/hda1   /boot          ext3    defaults              1 2
/dev/hda5   /              ext3    defaults              1 1
/dev/hda6   swap           swap    defaults              0 0
home:/home  /home                  nfs          defaults 0 0
/dev/sda1   /mnt/pendrive  vfat    user,noauto,umask=000 0 0
/dev/fd0    /mnt/floppy    vfat    user,noauto,umask=000 0 0
/dev/cdrom  /mnt/cdrom     iso9660 ro,user,noauto        0 0
```

Ha ezekkel megvagyunk, érdemes saját kernelt fordítani, a gyári kernel ugyanis nem mindig elég. Amiket célszerű belefordítani:

- Támogatást az összes létező lokális fájlrendszerhez
- A hálózati fájlrendszerekből érdemes az *NFS*-t és a *Samba*-t
- Az összes kódlapot
- *USB EHCI*-t (alias *USB 2.0*) és *USB OHCI*-t vagy *UHCI*-t, igény esetén mindkettőt
- *USB HID*-t, hogy ne jöjjön zavarba a masina egy *USB*-s egér esetén
- *SATA* támogatás (ez a 2.6-os kernelben a *SCSI* alrendszer alatt található)
- illetve jó benne van az összes hálókártya, ki tudja, mikor kell valamelyikben cserélni...

Amennyiben többféle gyártótól származó alaplapunk van, érdemes az összes alaplap vezérlőjét és hangkártya vezérlőjét befordítani – később majd látni fogjuk, miért kell ez. Ha esetleg több különböző generációs gépünk van – például *Pentium III* és *Pentium 4* – úgy érdemes a gyengébbre optimalizált kernelt készíteni. Ha két külön kernelt szeretnénk, akkor később lesz még dolgunk, de ne szaladjunk előre. Tegyük fel, hogy lefordult a kernel. Amennyiben *Nvidia* kártyánk van, úgy érdemes az *Nvidia* meghajtót használni. Ennek csak egy hátránya van: amikor új kernelt fordítunk, le kell azt is újra fordítani.

Klónozás, avagy legyen sok hasonló gépünk

Mostanra tehát egy jól működő *Linux* asztali géppel kell rendelkeznünk. Ha ez megvan, jöhet a klónozás. Tegyük fel, hogy egy *Pentium III*-as géppel végeztünk, de a gépparkban van jónéhány *Pentium 4*-es is. Tegyük a két előkészített *Knoppix CD*-t a két masinába, és bootoljuk be. A *Knoppix* indításakor érdemes `knoppix 2`-vel indítani, így nem indul el a grafikus felület. Indítsuk el a *Pentium 4*-en az ssh szervert a

```
/etc/init.d/ssh start
```

paranccsal, majd hozzunk létre egy a korábbi partíciós táblához hasonlót a *cfdisk* paranccsal.
Hozzunk létre a */mnt* könyvtárban két csatolási pontot, mondjuk *klon1* és *klon5* néven. A *Knoppix root* jelszavát is meg kell változtatni, legyen ez mondjuk *„knoppix”*.
A másik (eredeti) gépen is bootoljunk *Knoppixot* a már ismert módon. Csatoljuk be a */mnt/hda1* és */mnt/hda5* partíciókat. Ha ezzel megvagyunk, elkezdhetjük másolni a fájlrendszert az alábbi parancsokkal (a forrás gépen kell kiadni a parancsokat).

```
cd /mnt/hda1 && tar -czf - ./ | ssh root@celgep
➥'tar -xzpf - -C /mnt/klon1'
cd /mnt/hda5 && tar -czf - ./ | ssh root@celgep
➥'tar -xzpf - -C /mnt/klon5'
```

Az első parancs a */boot* partíciót másolja, míg a másik a root partíciót másolja. Ez jó ideig eltart. Az igazi profik az ilyesmit az ebédszünetre időzítik. Ha kész, és nem írt a tar semmi hibát, úgy indítsuk újra a célgépet. Ha rendben elindul, és az összes perifériát látja, akkor jól csináltunk mindent. Tegyük fel azonban, hogy nem. Makacsul csak annyit ír ki, hogy *GRUB*. Ekkor a célgépet bootoljuk újra a *Knoppix CD*-ről a már ismert módon. Indítsuk el a *GRUB*-ot.

```
install (hd0,0)/grub/stage1 (hd0)
➥(hd0,0)/grub/stage2 p (hd0,0)/grub/menu.lst
```

Ezzel a *GRUB* bekerül a jól megérdemelt helyére, és a következő indításkor megfelelően fog indulni.
Ezután a másolt gép, amelyről feltettük, hogy jobb processzorral bír, mint az eredeti, is nekiállhat kernelt fordítani, nem szabad azonban elfeledkeznünk, hogy amennyiben a gyári *Nvidia* meghajtót használjuk, úgy azt is újra kell fordítani. A nagyüzemi klónozás előtt érdemes megnézni, hogy az alaplap *DMA*-ja be van-e kapcsolva (be van-e fordítva a kernelbe), ugyanis nem mindegy, hogy a merevlemez 3 megabyteot vagy 30 megabyteot olvas másodpercenként. Érdemes még az *ssh-host* kulcsokat is törölni (*/etc/ssh*

Amennyiben fontosnak tartjuk a hallgatókat tájékoztatni még a belépés előtt az üzemeltetéssel kapcsolatos dolgokról (például nyitva tartás, szoftverek, stb.), úgy érdekes lehet a ***xmhtml widget set*** demóprogramja, ami egy egyszerű webböngésző. Némi forráskód átírás után elérhető (az egyik kollégám megcsinálta), hogy a program ne tudjon kimenni az internetre és még akármilyen filet ne is tudjon megnyitni. Ahhoz hogy ez működjön, egyelőre a ***KDM***-et kell használni. (már ígérik, hogy lesz a ***GDM***-nek is ilyen szolgáltatása)

Fedora alatt a ***kdmrc***-t kell megszerkeszteni, illetve az ***Xsetup*** fájlba beírni ezt a sort:

```
/home/tajek/info/browser
/home/tajek/info/index.html > /dev/null 2>&1 &
```

Debian esetén hasonlóan kell eljárni. Persze ez azt feltételezi, hogy az ***NFS*** meghajtón a `tajek` felhasználó könyvtárában megvannak a megfelelő fájlok.

könyvtárban), ugyanis a klónozás mellékhatásaként a 100-200 gépnek ugyanaz lenne a host kulcsa. Ez pedig nekünk értelemszerűen nem jó, meg úgy általában se igazán... Ezzel elérkeztünk nagyüzemi klónozáshoz.
Mindkét mintagépünk partíciós tábláját lementjük mondjuk a pendrive-unkra:

```
sfdisk -d /dev/hda > /mnt/pendrive/p3part
sfdisk -d /dev/hda > /mnt/pendrive/p4part
```

A merevlemez nem használt részeit ki kell nullázni, hogy a lemezképet (disk image) tartalmazó ***FTP*** szerveren kevesebb helyet foglaljon a betömörített képfájl. A kinullázás így történik:

```
dd if=/dev/zero of=/zeros && rm /zeros
```

Ha ezzel megvagyunk, helyezzük be indulásként az első (***Pentium III***-as) gépbe a ***g4u***-t tartalmazó floppyt vagy ***CD***-t. Ha befejezte a bootolást, adjuk ki az alábbi parancsot:

```
uploadpart 192.168.1.254 p3boot.tar.gz wd0e
uploadpart 192.168.1.254 p3linux.tar.gz wd0i
```

Ekkor a ***g4u*** verziójától függően pöttyök jelennek meg, vagy elkezdi számolni a megabájtokat. Ilyenkor a floppyt/CD-t nyugodtan kivehetjük a masinából és a másik gép következik:

```
uploadpart 192.168.1.254 p4boot.tar.gz wd0e
uploadpart 192.168.1.254 p4linux.tar.gz wd0i
```

Fontos, hogy ennél a műveletnél min. 30-40 gigabájtos legyen az FTP szerverünk üres tárhelye, ki tudja, mekkorák lesznek az képfájlok. Most jön a munka lélekölő része.
A többi gépen egyenként végre kell hajtani: a ***Knoppix*** bootolása becsatoljuk a pendrive-ot, és begépeljük ezt:

```
sfdisk /dev/hda < /mnt/pendrive/p3part
```

vagy

```
sfdisk /dev/hda < /mnt/pendrive/p4part
```

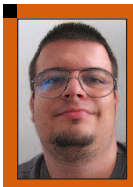
Ezzel a mintagépeken lévő partíciós táblát sikerül elővarázsolni. Ezután az összes gépen be kell bootolni a ***g4u***-t, és letölteni az FTP kiszolgálóról a megfelelő képfájlokat:

```
slurppart 192.168.1.254 p4boot.tar.gz wd0e
```

A ***g4u***-ról bővebb információt kaphatunk a parancssor fölötti részen.
Miután gépenként letöltöttük a két lemezrészt (a ***boot*** és a ***root***), még a ***GRUB***-ot is be kell üzemelni a már ismert módon. Ezzel lényegében készen vagyunk. Azzal még finomíthatunk az adminisztrációs részen, hogy a ***root*** felhasználónak generálunk kulcsot, és ezzel a ***root*** jelszó begépelése nélkül lehet ügyködni a gépeken. Aztán ott a másik megoldás, a ***pamusb***, amivel elérhető az is, hogy csak egy megfelelő titkos kulccsal ellátott ***USB kulcstartó (pendrive)*** társaságában működjön a ***root*** jelszó az adott gépen.

Végszó

Velem egyszer már előfordult, hogy a mintagép elszáll telepítés közben. Ilyenkor sajnos kezdhetünk memóriahibára gyanakodni és célszerű a Memtest nevű segédprogramot a gépen 1-2 napig futtatni, bár ez egyébként is erősen ajánlott új gépek esetén...



Medve Zoltán

2001-ben kezdett „Linuxolni”, de már korábban is ismerkedett a szabad szoftverek világával. Ha éppen nem a gép előtt ül, akkor fotózgat, olvasgat vagy bicajozik.

KAPCSOLÓDÓ CÍMEK

NFS
➲ http://www.tldp.org/HOWTO/NFS-HOWTO/index.html
LDAP
➲ http://www.tldp.org/HOWTO/LDAP-HOWTO/index.html
DHCP
➲ http://www.tldp.org/HOWTO/DHCP/index.html
g4u
➲ http://www.feyrer.de/g4u/
Knoppix
➲ http://www.knopper.net/knoppix-mirrors/index-en.html
memtest86
➲ http://www.memtest86.com/
pamusb
➲ http://www.pamusb.org/

Mi folyik a háttérben – GkrellM

A GkrellM egy kis méretű, és funkciókban igen gazdag, grafikus rendszermonitorozó program. Egyik óriási előnye, hogy kevés erőforrást igényel.

Ennek köszönhetően más rendszermonitorozó programokkal szemben mint a *gdesklets* (*Gnome*) vagy a *superkaramba* (*KDE*), gond nélkül használható egy kisebb teljesítményű számítógépen például *Fluxbox* ablakkezelő vagy *Xfce* asztali környezet alatt is. Segítségével mindig szem előtt tarthatjuk a processzor terhelését, a hálózati ki-; és bemenő forgalmat, a merevlemez(ek) telítettségét vagy akár újonnan érkezett leveleink számát. Formájából adódóan kevés területet vesz el a helyenként amúgy is kihasználatlan oldalsávból és egy jól megválasztott bőrrel (skinnel) szinte beleolvad az asztalba (vagy a panel stílus közé). Használatba vételével hamar beláthatjuk, hogy egy jól behangolt rendszermonitor hatalmas segítség lehet számunkra munka közben.

A programnak *gtk 2.0*, *gdk 2.0*, *glib 2.0* függőségei vannak, bővebb felvilágosítást az *INSTALL* fájl nyújt. *Linux* rendszereken a */proc* fájlrendszerből vett információk alapján dolgozik. A projekt tulajdonosa és főfejlesztője *Bill Wilson*. Érdemes megjegyezni, hogy mint sok szabad szoftvernek, ennek is létezik más platformokon futtatható változatai, így elérhető *FreeBSD*, *NetBSD*, *OpenBSD*, *MacOS X*, *Solaris* és *Windows* rendszereken is.

Első hivatalos kiadása 1999. júliusában jelent meg. Ekkor 0.5 volt a verziószáma, de csak a 0.6.0-ás kiadásnál vette fel a projekt a *GkrellM (The GNU Krell Monitors)* nevet. Jelenleg a legfrissebb stabil kiadása a 2.2.7-es.

Beszerzés

A magyar terjesztések (*BlackPanther*, *Frugalware*, *UHU*) tartalmazzák csomag vagy forrás formájában, illetve minden nagyobb kiadáshoz megtalálható előre gyártott csomagban. Ha mégsem szerepel a csomaglistánkban, akkor a ➲ *www.gkrellm.net* oldalról lehet letölteni tömörített forrás, valamint bináris csomag formájában (*Debian*, *Mandrake*, *RedHat*, *Slackware* és *SuSE* terjesztésekhez, illetve a **BSD* kiadásokhoz). A programokhoz további skineket a *http://www.muhri.net/gkrellm/* honlapról szerezjetünk be. (Az ábrákon az *Egan* és az *Invisible skin* látható). Ezeket a */home/userneve/.gkrellm2/themes* könyvtárba kell másolni és kitömöríteni.

Telepítés után terminálról a `gkrellm` parancs kiadásával indítható, célszerű néhány alapbeállítást ekkor elvégezni. A menü egy a fejlécen történő jobb kattintással hívható elő (mozgatni a bal gombbal lehet, de csak a fejléc körül megragadva). Érdemes átfutni a program súgójában (terminálon `man gkrellm`), mert igen részletes információkkal szolgál. Például ha minden induláskor szeretnénk, hogy induljon és ugyanott jelenjen meg, de nincs lehetőségünk ezt beállítani a grafikus felületen, akkor a következőt kell tennünk: létrehozunk egy fájlt a *~/Desktop/Autostart* könyvtárban, futtathatóvá tesszük (`chmod +x <fájl neve>`) majd valamilyen szövegszerkesztővel felvesszük bele a következő sorokat:

```
#/bin/bash
gkrellm  -g +960+30
```

Ennek hatására minden alkalommal, amikor a grafikus felület elindul a gkrellm elindul és a vízszintes 960, függőleges 30 képpontra pozicionál. Elindult a program, mit is látunk itt voltaképpen? Egy függőleges téglalap alakú panelt, melynek a legfelső sorban szerepel a rendszer neve valamint a kernel verzió, alatta a pontos idő. Ezt követően a rendszert monitorozó diagramok/ledek (*CPU*, hálózat, folyamatok), majd memória és swap állapota látható. Ezt követik a partíciók és különböző adathordozók állapota (be van-e csatolva) és telítettsége, majd a levél figyelő és végül a bővítmények (plugin-ek) kapnak helyet. Legalul pedig az *uptime* vagyis a bootolás óta eltelt idő. A lényeges, illetve a működéssel kapcsolatos beállítások menüből végezhetők, de a monitorok megjelenésére vonatkozó dolgok, mint például a kijelzés módja, a segédvonalak, az ablak magassága az adott kijelzőn történő jobb kattintással előugró menüben állítható.

Beállítás

Az alapbeállítások (*General*) között szerepel, hogy emlékezzen-e a következő indításnál a pozícióra, megjelenjen-e a tálcán valamint a munkaasztalokon (illetve azok miniatűrjein), milyen gyakran frissítse a kijelzőket és mekkora szélességgel jelenjen meg. A beépített funkciók (*Builtins*) közül nem említeném meg az összeset, ezek felfedezését az olvasóra bízom, viszont kiemelnék néhányat, melyek nagyban megkönnyítik a felhasználó munkáját. Monitorok illetve ledek közé *CPU*, *Proc*, *Disk*, *Internet* és *Net* eszközök tartoznak, melyek működését illetve teljesítményét egy diagramon követhetjük szemmel. A *Net* ablakon a jobb alsó sarokba kattintva egy kis felugró ablak tájékoztat a napi, heti és havi ki-; bemenő adatforgalom mennyiségéről. Az ablak alatt meg lehet jelentetni az internet csatlakozástól eltelt időt, mely szintén segítség egyrészt a modemmel internetezők számára, másrészt azoknak akiknél a szolgáltató 24 óránként kapcsolatot bont.

A *File System* lényeges és hasznos eszköz, segítségével egyrészt látható az adott média/partíció telítettsége, másrészt ha nem használunk semmiféle automatikus média-csatoló eszközt (*automount*/*supermount*/*submount*), akkor kényelmesen egyetlen kattintással befűzhetjük a kívánt médiát. Optikai meghajtó esetén még ki is nyittathatjuk lecsatoláskor a tálcát. Sőt, a kijelzőre kattintva fényújság formájában kijelzi az aktuális állapotot (alapértelmezés szerint a teljes kapacitás/szabad hely).

A *Mail* segítségével szemmel tarthatjuk állandóan postafiókunk vagy állapotát, helyi (*local*) és távoli (*remote*) fiókokat egyaránt, utóbbi esetben a szerver nevét, protokoll típusát (*IMAP*/*POP3*), felhasználónevet valamint jelszót kell megadni, illetve azt, hogy milyen gyakorisággal ellenőrizze a postafiók állapotát. Sőt ha megadjuk a program nevét akkor, gombnyomásra megnyitja kedvenc levelező programunkat is.

Bővítmények

A funkciók kétféle módon bővíthetőek: csomag illetve forrás formájában. Előbbi esetben nincs sok tennivaló, le kell tölteni és fel kell telepíteni, majd újraindítani a programot, majd a *Plugin* menüpontban megjelenik a telepített bővítmény, ahol engedélyezni kell és használatba is vehető. Előfordul azonban, hogy csak forrás formájában tudunk beszerezni bizonyos bővítményeket. Semmi gond, általában kis méretű programokról van szó, és a *readme* fájlok igen részletes útmutatást adnak. Általában csak egy a forrás könyvtárában kiadott `make` parancs szükséges, mely hatására létrejön egy *.so* kiterjesztésű állomány, melyet be kell másolni a *~/.gkrellm2/plugins/* vagy a */usr/lib/gkrellm/plugins* könyvtárba (előbbi esetben természetesen csak az adott felhasználó fogja látni).

A népszerű bővítmények közül kiemelném a *Gkrellmms*, *gkrellmlaunch* valamit a *gkrellshoot* nevűeket.

A *Gkrellmms* egy *xmms* plugin, mely segítségével a képeken látható multimédia-vezérlő gombok kerülnek a panelra, valamint olvashatjuk az éppen lejátszott dal címét. Praktikus megoldás, ha több munkaasztallal dolgozunk és számot akarunk váltani, vagy csak egyszerűen kényelmetlen az *xmms* ablakot állandóan elővarázsolni. A *gkrellmlaunch* segítségével program indító gombokat helyezhetünk a *GkrellM* paneljára.

A *gkrellshoot* kettős funkciót lát el, egyrészt egy kattintással készíthetünk felvételt az asztalról vagy egy-egy ablak tartalmáról, másrészt zárolhatjuk a képernyőt és mintegy extra funkcióként még miniatűr animációkat is megjeleníthetünk. További hasznos eszköz a *Volume plugin*, mely a különböző csatornák (*Master*, *CD*, *PCM*, mikrofon...) hangerejét szabályozza (természetesen itt is lehetőségünk van testre szabni, mely csatornák jelenjenek meg a panelon).

A program felhasználói fájljai a *~/.gkrellm2* könyvtárban találhatóak. Itt a *data* könyvtárban található a startpozíció, a *plugins* és *themes* könyvtár neve fedi a tartalmát, ezeknek akkor van fontos szerepük, ha nem rendelkezünk rendszergazdai (root) jogosultsággal az adott számítógépen, de szeretnénk egyéni skineket használni vagy saját plugint hozzáadni. Ezen kívül néhány fájl található még itt: a *plugin_enable* tartalmazza az engedélyezett bővítményeket, a *theme_config* az aktuális skint valamint a betűtípusokat és végül az *user-config* tartalmaz minden beállításra vonatkozó információt (egy esetleges rendszer csere utáni visszaállításhoz ezt célszerű lementeni, majd „utánhangolni").

Remélem felkeltettem az érdeklődést eme nagyszerű program iránt, és hasznos információs forrást találtok benne.

Rácz Zoltán
(razoli@linuxforum.hu)
Jelenleg egyetemista az ELTE informatika-matematika tanári szakán. A Linuxszal két éve került kapcsolatba az UHU 1.0 kapcsán. Fél éve egyetlen operációs rendszer van a gépén: egy Debian Sid.

Gyorshajtás 3D-ben

nVidia és ATI grafikus kártyák telepítése lépésről lépésre

Grafikus meghajtót általában könnyű Linux alá telepíteni. A VGA, illetve a VESA meghajtók még az egzotikus videókártyákkal is működnek. A gond akkor kezdődik, ha olyan játékot, illetve programot akarunk futtatni, amely 3D gyorsítást igényel. Ennek a beállításáról ugyanis már nekünk kell gondoskodni...

Ebben a cikkben azt mutatom be, hogyan telepíthetjük a két legelterjedtebb 3D gyorsítással rendelkező kártyatípus (***nVidia*** és ***ATI***) vezérlőit néhány különböző ***Linux*** terjesztés alatt. A telepítést ***nVidia FX5200*** és ***ATI Radeon 9600 RV350*** kártyákkal végeztem. Előre kell bocsátanom, hogy a leírtak használatához élő internet kapcsolatra lesz szükség. A tesztelés során a ***Tuxracer (PlanetPenguin Racer)*** nevű játék volt segítségemre, mivel ez egyrészt 3D gyorsítást igényel, másrészt valamennyi általam ismertetett terjesztésnek része. (A játék elindul ugyan 3D gyorsítás támogatása nélkül is, de lassú, és a szaggatott képernyőképek miatt nem élvezhető.)

Ubuntu Linux – előkészületek

A terjesztés jelenleg a legfrissebb változata az ***Ubuntu „Hoary" 5.04*** verzió. A terjesztésben a csomagok három csoportba sorolhatók: ***restricted***, ***universe*** és ***multiverse***. A telepítés után alapértelmezés szerint csak a restricted csoport érhető el. Mivel szükségem volt a másik két csoport néhány csomagjára is (a ***Tuxracer*** például a ***universe*** csoportban tartozik), először azokat is elérhetővé kellett tennem. A műveletek végrehajtásához root jogosultságra van szükség, ezért ***root terminált*** indítottam: ***Alkalmazások -> Rendszereszközök -> Root terminál***

A ***nano*** szövegszerkesztővel módosítottam a ***/etc/apt/sources.list*** fájl tartalmát (lásd a dobozban):

```
nano /etc/apt/sources.list
```

Ezután előbb csomagok listáját, majd magukat a csomagokat is frissítettem:

```
apt-get update
apt-get upgrade
```

Végül a ***Tuxracert*** a következő paranccsal telepíthetjük:

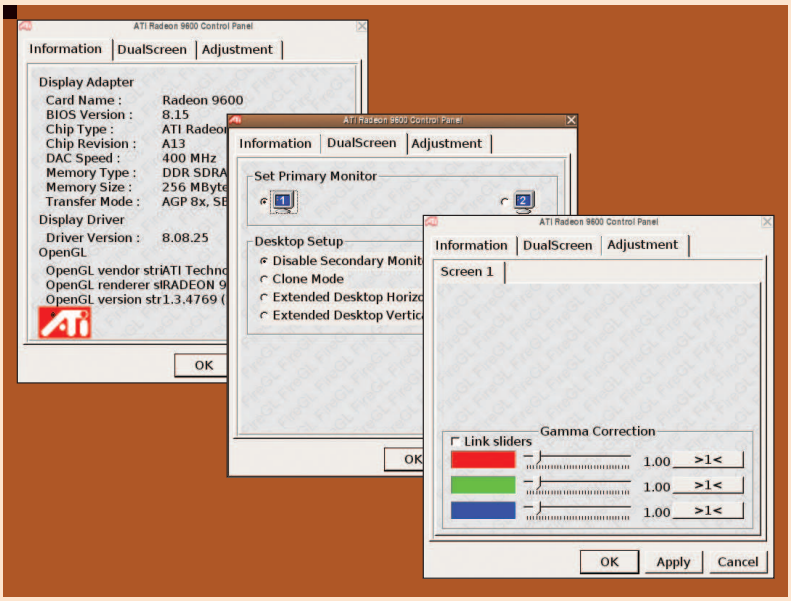
```
apt-get install tuxracer
```

/etc/apt/sources.list:

```
deb http://hu.archive.ubuntu.com/ubuntu hoary main restricted
➥universe multiverse
deb http://hu.archive.ubuntu.com/ubuntu hoary-updates main
➥restricted universe multiverse
```

/etc/X11/xorg.conf (részlet):

```
Section "Device"
  Identifier "ATI Technologies, Inc. Radeon 9600 (R300 AP)"
# Driver           "ati"
  Driver           "fglrx"
  BusID            "PCI:1:0:0"
EndSection
```



Az ATI meghajtó telepítése Ubuntu Linux alatt

Először telepítettem az *ATI GL* és *GLX* támogatást is tartalmazó meghajtójának csomagját:

```
apt-get install xorg-driver-
➥fglrx
```

Ezután a már említett *nano* szövegszerkesztő segítségével megfelelően módosítottam az X szerver beállító fájlját:

```
nano /etc/X11/xorg.conf
```

A meghajtó eredeti típusát (`Driver „ati”`) beállító sor elé megjegyzésjelet tettem („kikommenteztem”) és beszúrtam egy új sort ami a megfelelő meghajtó használatára utasítja a rendszert (`Driver „fglrx”`) bővítettem (lásd a dobozban). A változáso érvénybe lépéséhez újra kell indítanunk a rendszert (***Rendszer->Kijelentkezés-> Újraindítás OK***).

Nem kötelező ugyan, de én telepítettem még a *fglrx-control* csomagot is, mellyel szükség esetén az *ATI* kártya beállításait lehet módosítani:

```
apt-get install fglrx-control
```

Telepítés után ezt az eszközt (lásd a lap tetején a képeket) a `fireglcontrol` paranccsal indíthatjuk el (például az *Alkalmazások ->Alkalmazások futtatása...* menüpontban).

nVidia meghajtó telepítése Ubuntu Linux alatt

A műveletek végrehajtásához rendszergazdai jogosultságra van szükség, ezért *root terminált* indítunk:

Alkalmazások -> Rendszereszközök -> Root terminál

Telepítjük az *nVidia 3D* gyorsítást támogató meghajtójának csomagját:

```
apt-get install nvidia-glx
```

A */etc/X11/xorg.conf* fájlt most nem kell kézzel átírnunk. A beállítások automatikus módosításához csupán a következő parancsot kell kiadnunk:

```
nvidia-glx-config enable
```

Megint nem kötelező, de telepíthetjük az *nVIDIA* beállítóprogramját is:

```
apt-get install nvidia-settings
```

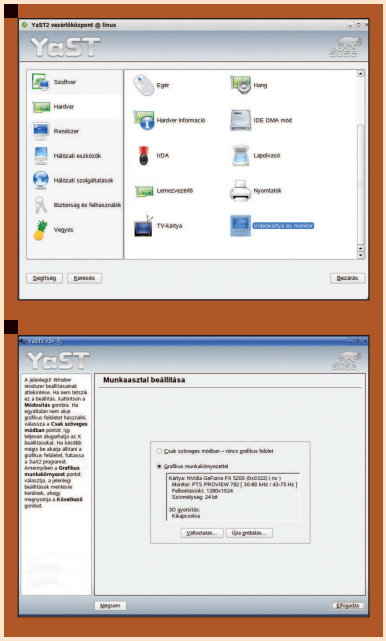
Ezt most egy kicsit más helyen találjuk, az `nvidia-settings` parancsot kell indítsuk (lásd a képeket a következő oldalon)

SUSE Linux

A *SUSE* jelenlegi legfrissebb változata a 9.3-as, így én is ezzel próbáltam ki a telepítést. Ebben a terjesztésben a *SAX2* programmal lehet az *X* kiszolgálót beállítani, ami a *YAST2* központi vezérlőprogramból is elindítható.

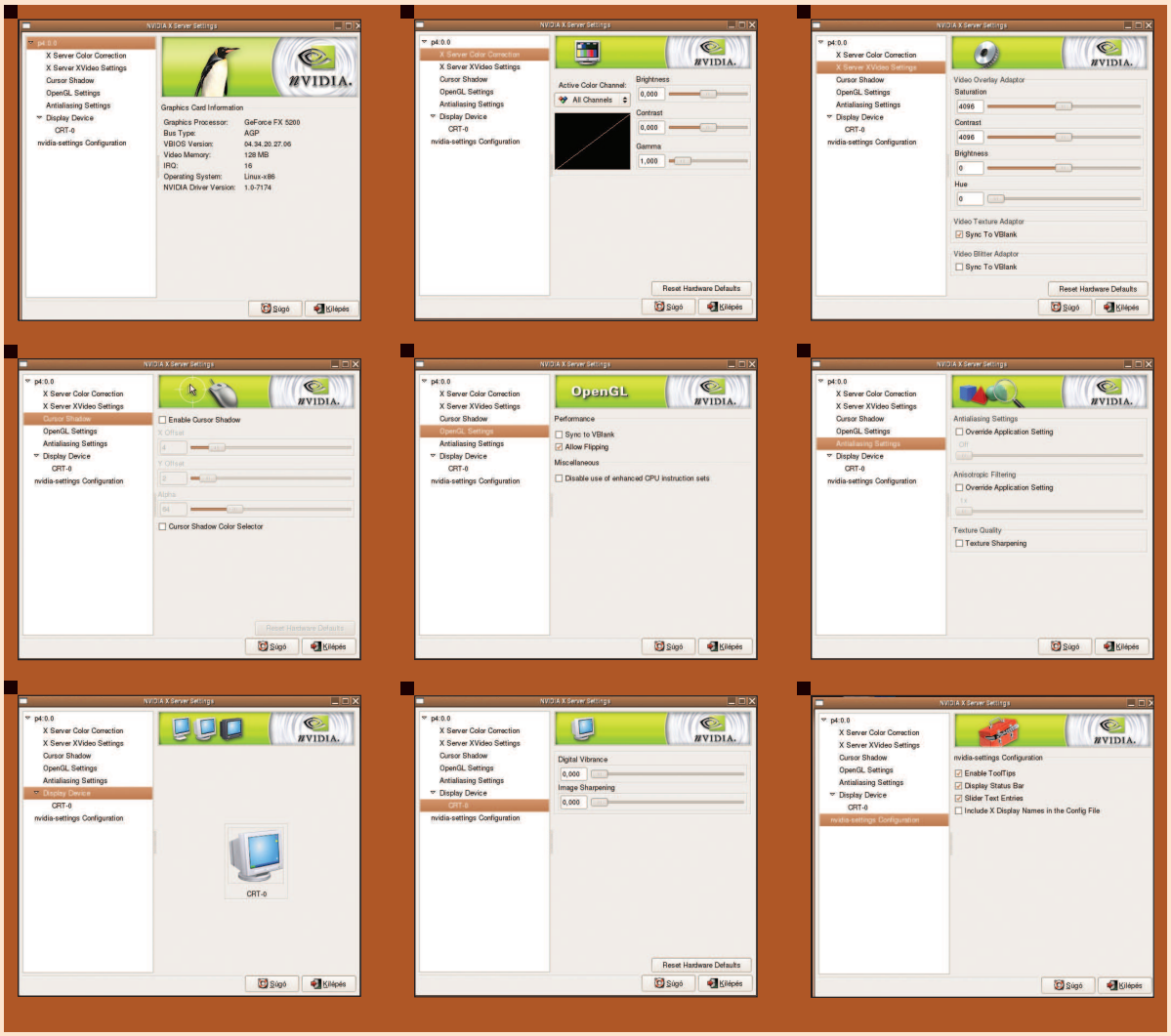
ATI meghajtó telepítése SUSE Linux alatt

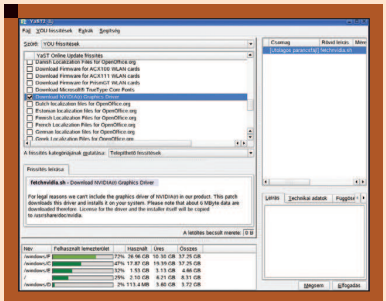
Kicsit meglepő, hogy a 3D gyorsítást támogató *ATI* meghajtót a *SUSE 9.3* eleve tartalmazza, a *SAX2* segítségével mégsem lehet azt beállítani. Ezért *root terminált* indítottam, majd az `fglrxconfig` parancs segítségével végeztem el a szükséges módosításokat. Ez egy karakteres beállítóprogram, mely kérdéseket tesz fel az egérrel, a billentyűvel a monitorvezérlő beállításaival és a monitorral kapcsolatban. Ha pedig mindenre megkapta a választ, akkor elkészíti, illetve automatikusan módosítja a */etc/X11/xorg.conf* fájlt. A változások érvénybe léptetéséhez természetesen itt is célszerű újraindítani a rendszert.

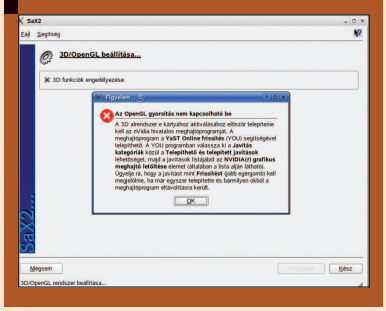


nVidia meghajtó telepítése SUSE Linux alatt

A *SUSE* az *nVidia 5200-as* kártyát (és általában az *nVidia* kártyáit) automatikusan felismeri, de a 3D gyorsítást alapértelmezés szerint nem kapcsolja be. Ezt nekünk kell engedélyezni. Mielőtt azonban ezt megtennénk, le kellett töltenünk az *nVidia* grafikus meghajtóprogramját, mivel a terjesztés ezt (feltehetőleg jogi problémák miatt) nem tartalmazza. (Ha valaki ezt a lépést elfelejtené, akkor nem tudja engedélyezni a 3D gyorsítást, csupán egy megfelelő figyelmeztető üzenetet kap.)
3D gyorsítás engedélyezését a *SUSE* központi beállítóprogramjával a következőképpen végezhetjük el:







YAST2 -> Hardver -> Video kártya és Monitor -> Változtatás -> Munkaasztal -> 3D gyorsítás -> Beállítások Módosítása -> 3D funkció engedélyezése

A változások most is a rendszer újraindítása után lépnek csak érvénybe. Érdemes megjegyezni, hogy az ***nVidia*** időről időre javítja a kártyáihoz használható meghajtóprogramot, így azt néha nem árt frissíteni. Ehhez a ***Yast*** vezérlőközpontban a következőket kell tennünk:

YAST -> Szoftver -> Online frissítés-> Download NVIDIA(r) Graphics DRiver

UHU Linux

A telepítést az ***UHU-Linux 1.2 Office*** változattal végeztem. Ez a terjesztés is felismeri automatikusan a monitorvezérlőt, de a 3D gyorsítás támogatását nem tartalmazza. A telepítéshez itt is rendszergazdai jogokkal kell rendelkeznünk, ezért a parancsokat ***root terminálban*** kell végrehajtani.

ATI meghajtó telepítése UHU Linux alatt

A meghajtót telepítő programot az ***ATI*** honlapjáról töltöttem le (lásd: *[kc1]*) Az anyagot helyezzük egy tetszőleges könyvtárba, majd futtassuk a telepítőprogramot, mely létrehozza a meghajtóprogramokat:

```
sh ./ati-driver-installer-
➥8.14.13.run
```

A program futásakor 3 ablak jelenik meg egymás után, ahol én az ablakokban rendre a következőket választottam:

1. ablak: *Options: Install Driver*
2. ablak: *Automatic*
3. ablak: *Exit*

Az X kiszolgáló automatikus beállításához itt is futtassuk a beállító programot és persze válaszoljunk a kérdéseire (ugyanúgy, mint a ***SUSE Linuxnál***), aztán indítsuk újra a rendszert:
`/usr/X11R6/bin/fglrxconfig`

A `fireglcontrol` program természetesen itt is rendelkezésünkre áll.

Indítása:
```
/usr/X11R6/bin/
➥fireglcontrolpanel
```

nVidia meghajtó telepítése UHU Linux alatt

A meghajtót telepítő programot (lásd: *[kc2]*) az ***nVidia*** honlapjáról töltöttem le (lásd: *[kc3]*).
A meghajtó egyes részeinek lefordításához szükség van az általunk használt kernel forrására, ezért az ezt tartalmazó csomagot is telepítenünk kell:
```
apt-get install kernel-source
```

A meghajtó lefordítása és telepítése előtt le kell állítani az ***X*** kiszolgálót. Én ilyenkor azt a trükköt használom, hogy az ***X*** kiszolgáló beállító fájljába (*/etc/X11/xorg.config*) egy tetszőleges szövegszerkesztővel beírom egy nem létező meghajtó nevét (például `Driver „kisnyul”`), majd ***CTRL+ALT+Backspace*** megnyomásával újraindítottam az ***X*** kiszolgálót. Ez aztán természetesen már nem tud elindulni. Így nézett ki tehát nálam a módosított beállító fájl:

```
Section "Device"
   Identifier   "UHU-Card"
   Driver       "kisnyul"
  EndSection
```

Lefordítottam és telepítettem az ***nVidia*** meghajtót (közben megkérdezte, hogy elfogadom-e a licencet, illetve hogy letöltse-e az ***nVidia*** honlapról, ha van újabb telepítőprogram):

```
sh NVIDIA-Linux-x86-1.0-7667-
➥pkg1.run
```

Végül szövegszerkesztővel módosítottam az ***X*** kiszolgáló beállító fájlját (*/etc/X11/xorg.config*) úgy, hogy a meghajtóprogram nevéhez `nvidia`-t írtam, aztán újraindítottam a rendszert:

```
Section "Device"
   Identifier   "UHU-Card"
   Driver       "nvidia"
EndSection
```

Debian Linux

A telepítést ***Debian Linux 3.1 „sarge”*** verzióval végeztem. Itt is le kellett fordítanom a meghajtókat, amihez a kernelforrás telepítésére is szükség volt. Mivel ***Debiant*** használok a gépemen, először fordítottam egy a saját igényeimnek megfelelő kernelt. Az ***ATI*** jelenleg letölthető telepítőprogramja (***ati-driver-installer-8.14.13.run***) a dokumentáció szerint a ***2.6.11.x*** verziószámú kernelekkel kompatibilis, így ezek közül a legújabbat választottam. A telepítés végrehajtásához természetesen itt is rendszergazdai jogosultság szükséges.

A 2.6.11.x rendszermag fordítás Debian Linux alatt

Először töltsük le a rendszermag forráskódját:

```
wget http://kernel.org/
➥pub/linux/
➥kernel/v2.6/
➥linux-2.6.11.12.tar.bz2
```

Ezt a */usr/src* könyvtárba kell kicsomagolni, de előzetesen telepítenünk kell a ***bzip2*** és ***libncurses5-dev*** csomagokat:

```
apt-get install bzip2
➥libncurses5-dev
cd /usr/scr
tar xvjf linux-
➥2.6.11.12.tar.bz2
```

Létrehozunk egy a kibontás során létrejött könyvtárra mutató ***linux*** nevű szimbolikus linket, majd belépünk ide:

```
ln -sfn linux-2.6.11.12 linux
cd linux
```

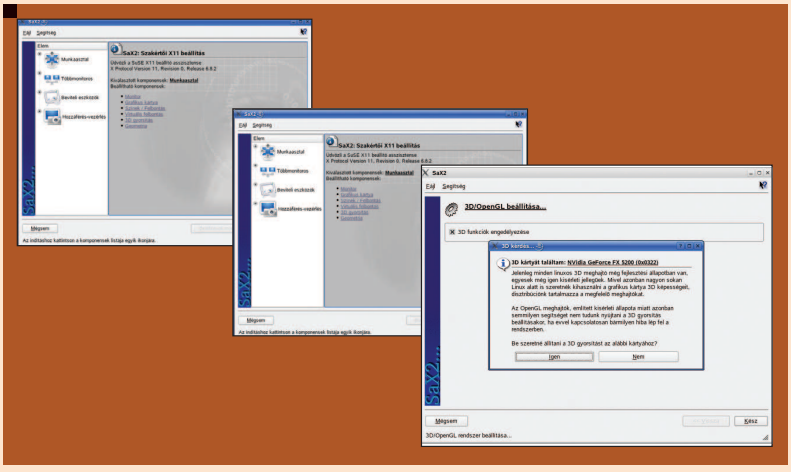
Indítsuk el a karakteres felületű, de menüvezérelt konfiguráló programot:
```
make menuconfig
```

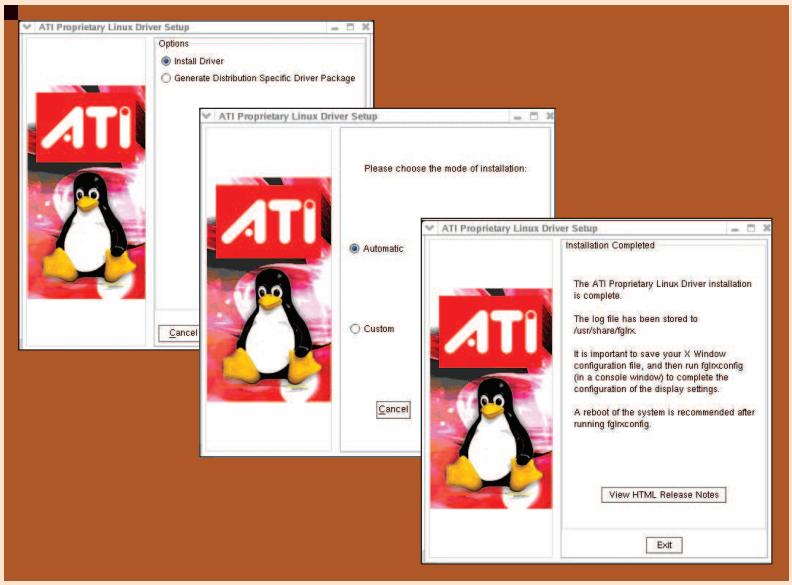
A következőket állítottam be (ha valaki nem tudná melyik opciót hol találja, az általam létrehozott konfigurációs fájlt le tudja tölteni a ***Linuxvilág*** honlapjáról).
Ha ez kész, akkor lefordítjuk az új rendszermagot és a hozzá tartozó modulokat:

```
make
make modules_install
```

Következnek az utolsó simítások... Bemásoljuk a rendszermagot, a konfigurációs fájlt és a rendszertérkép fájlokat a */**boot*** könyvtárba:

```
cp /usr/src/linux/arch/i386/
➥boot/bzImage /boot/
➥vmlinuz-2.6.11.12
cp /usr/src/linux/.config
➥/boot/config-2.6.11.12
cp /usr/src/linux/System.map
➥/boot/System.map-2.6.11.12
```





Létrehozzuk a rendszertöltéskor szükséges ***initrd*** képfájlt:

```
mkinitrd -o /boot/initrd.img-
➥2.6.11.12 2.6.11.12
```

Aktualizáljuk a modulok függőségi listáját, valamint a ***GRUB*** rendszertöltőt, végül újraindítjuk a rendszer:

```
update-modules
update-grub
```

Ha a rendszer újra felállt, biztos ami biztos alapon ellenőrizzük a kernelverziót:

```
uname -r
```

X kiszolgáló telepítése Debian Linuxra

A grafikus meghajtók telepítése előtt természetesen telepítenünk kell az ***X*** szervert és nem árt kipróbálni a működését is.

```
apt-get install xserver-xfree86
➥xutils xbase-clients
```

A telepítéskor a rendszer megkérdezte, hogy milyen monitorvezérlő van a gépemben (egyelőre ***VGA*** vagy ***VESA***), milyen az egér, a billentyűzet, valamint hogy milyen jellemzőkkel bír a monitor.
Ha minden rendben lefutott, akkor a kipróbáláshoz indítsuk el a grafikus felületet:

```
startx
```

1. kódlista

```
CONFIG_MODULES=y
CONFIG_MODULE_UNLOAD=y
CONFIG_KMOD=y
CONFIG_MTRR=y
CONFIG_AGP=m
CONFIG_AGP_ATI=m
CONFIG_TMPFS=y
CONFIG_DRM=m
# CONFIG_DRM_TDFX is not set
# CONFIG_DRM_GAMMA is not set
CONFIG_DRM_R128=m
CONFIG_DRM_RADEON=m
# CONFIG_DRM_I810 is not set
# CONFIG_DRM_I830 is not set
# CONFIG_DRM_I915 is not set
# CONFIG_DRM_MGA is not set
# CONFIG_DRM_SIS is not set
```

ATI meghajtó telepítése Debian Linux alatt

Debian alatt ugyanúgy telepíthetjük az ***ATI*** meghajtót, mint az ***UHU Linux*** esetében. Én is csupán annyit módosítottam a folyamaton, hogy most karakteres felületen futtattam a meghajtó telepítőprogramját:

```
sh ./ati-driver-installer-
➥8.14.13.run
```

Az ***X*** kiszolgál konfigurációs fájljának (***/etc/X11/XF86Config-4***) átírása is pontosan ugyanúgy történik, mint az ***UHU Linuxnál***:

```
/usr/X11R6/bin/
➥fglrxconfig
```

A fireglcontrol programot (ha telepítettük) itt a

```
/usr/X11R6/bin/
➥fireglcontrolpanel
```

paranccsal indíthatjuk.

nVidia meghajtó telepítése Debian Linux alatt

A meghajtót telepítő programot (lásd: [kc2]) az ***nVidia*** honlapjáról tölthetjük le a már ismert módon, sőt minden egyéb részlet is teljesen megegyezik az ***UHU Linuxnál*** elmondottakkal. Tekintettel arra, hogy minden terjesztésnél kicsit más módon sikerült eredményre jutnom, valószínű, hogy az itt nem tárgyalt Linux változatok többségénél is használható valamelyik bemutatott eljárás.

Sok sikert mindenkinek!

Jászberényi József

Szeret biciklizni, kirándulni, olvasni, sörözni és szabadban főzni. A stratégiai játékoktól a műszaki CAD programokig sok minden érdekli. Legtöbbet szerverprogramokkal foglalkozik és néha mérgelődik.
(jaszberenyij@pattantyus-gyor.sulinet.hu)

KAPCSOLÓDÓ CÍMEK

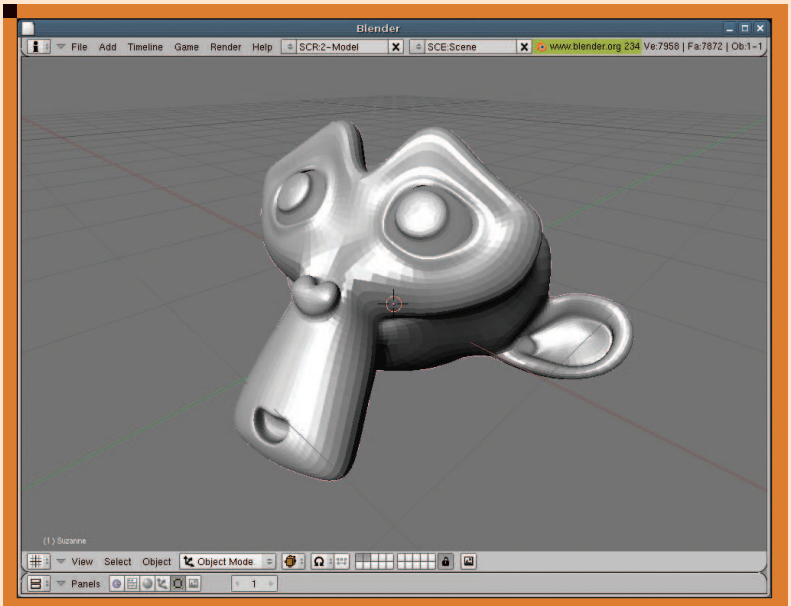
[kc1] ➲ http://www2.ati.com/drivers/linux/ati-driver-installer-8.14.13.run
[kc2] ➲ NVIDIA-Linux-x86-1.0-7667-pkg1.run
[kc3] ➲ http://www.nvidia.com -> Download Drivers -> Linux, FreeBSD, and Solaris Drivers->Linux IA32

Térhatás – a Blender használata

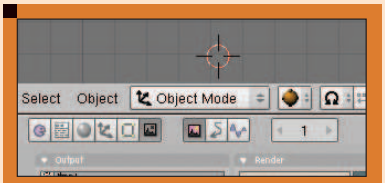
Bemelegítés

A Blender immár majdnem 10 éve folyamatosan fejlődő (Az 1.0-ás verziót 1996-ban adták ki), ingyenesen hozzáférhető 3D szerkesztő és renderelő alkalmazás. Linuxos környezetben ugyanúgy használható, mint Windows vagy BSD alatt. Sőt – és ez számomra a legmeglepőbb volt – egy egyszerű PDA-n is képes futni.

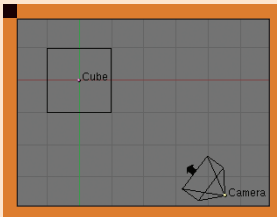
■ 2002 óta a *GNU General Public License (GPL)* alatt adják ki. Tudását szemlélteti, hogy míg a program maga alig 5-6Mb, a hozzá tartozó dokumentáció körülbelül 60Mb tömörítve.
A beépített ***Python*** programozási függvénykönyvtár segítségével a program hihetetlenül bővíthető, teljes mértékben átszabható. Már az alaptelepítés is tartalmaz néhány ilyen szkriptet, amelyekkel különböző formátumokba exportálhatunk, így az itt készített modelleket más programokban is felhasználhatjuk. Pillanatnyilag a 2.37-es verzió a legújabb, így a leírtak is erre vonatkoznak.
A felhasználói felület különböző panelekre tagolódik. Ezeket mi szabadon átméretezhetjük, kettéoszthatjuk, bezárhatjuk, stb., így bárki kialakíthatja a neki megfelelő környezetet, és természetesen el is mentheti azt. Minden egyes panel típusát beállíthatjuk, ami meghatározza mi jelenik meg az adott panelon. Létezik ***Buttons Window***, ***3D View***, ***Image Browser***, ***OOPS Schematic***, és sok egyéb más típus. Nekünk egyelőre csak a középső ***3D View*** ablakra lesz szükségünk, úgyhogy meg is foghatjuk az egérrel, és megnövelhetjük lefelé, amitől alul a gombokat tartalmazó panel szépen eltünedezik, de azt most úgysem fogjuk használni.
Van tehát egy majdnem teljes képernyőt elfoglaló 3D nézetünk. Ezen fogunk megtanulni dolgozni, ebből is legelőször csak ***Object Mode***-ban, amit az adott panel fejlécén tudunk állítani. Ha már ***Object Mode***-ban vagyunk (egyébként indításkor ez az alapméretezett), akkor hagyjuk is úgy. Nem kell meglepődni, ha az előbb említett „fejléc” alul van, nem pedig felül. Ez is egy állítható tulajdonság. Akinek úgy kényelmesebb, áthelyezheti: jobb



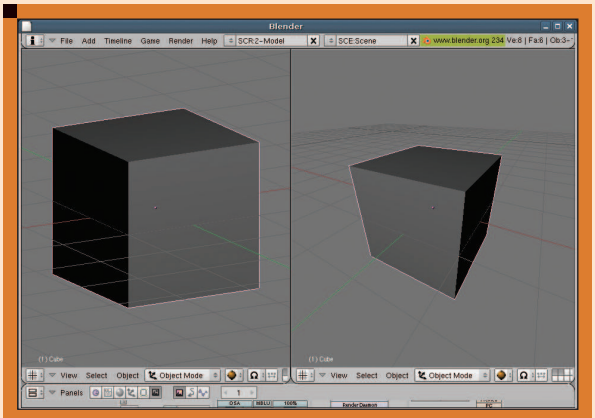
■ ***1. ábra*** A Blender 3D View ablaka



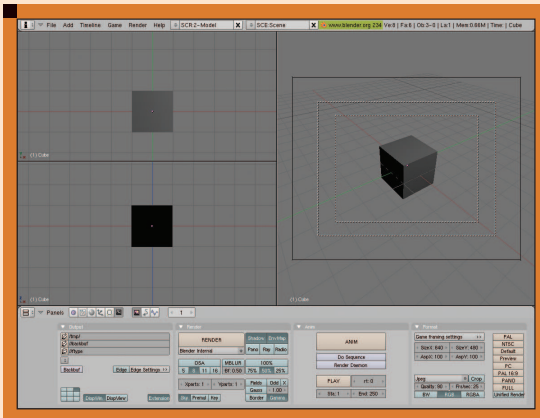
■ ***2. ábra*** Itt tudjuk beállítani a 3D View működési módját



■ ***3. ábra*** A kockánk felülről és a kamera



4. ábra Párhuzamos (bal oldal) és perspektivikus (jobb oldal) nézet



5. ábra Kockánk néhány különböző irányból

klikk a fejlécen, majd *„Top”*, *„Bottom”*, vagy *„No Header”*, attól függően, hogy felülre vagy alulra szeretnénk helyezni, esetleg el akarjuk tüntetni. Ez utóbbi azonban most nem ajánlott. Ha eddig ellenálltunk a kísértésnek, és nem kezdtünk el „eszeveszetten kattintgatni”, akkor még mindig csak egy kocka van előttünk felülnézetből, és egy másik furcsa, fekete alakzat kicsit balra lent. Az ott a „render kamera”. A program szempontjából a kamera is egy ugyanolyan objektum, mint barmi más, ugyanúgy tudjuk a térben forgatni, mozgatni, sőt akár törölhetjük is.

Az objektumok egymástól különálló testek, amiket szükség esetén ***összeilleszthetünk (Join)***, ***törölhetünk (Erase)***, vagy tetszőlegesen átszerkeszthetünk. Minden objektum rendelkezik egy köépponttal, ami egy sárga vagy rózsaszínű pont valahol az objektum belsejében. Jelen esetben a kockánknak pont középen van. Ahhoz, hogy ebben a 3D-s térben megfelelően tudjunk dolgozni, először meg kell tanulnunk több irányból látni az előttünk lévő virtuális modellt. Erre a legegyszerűbb mód a numerikus billentyűzet használata. A 4, 6, 8 és 2 billentyűkkel tetszőlegesen forgathatjuk a képet, a 7, 1, 3 és 0 gombok pedig előre beállított nézeteket adnak:

7 – Felülnézet
1 – Elölnézet
3 – Oldalnézet jobbról
0 – A kamera nézőpontja

A - és + -al távolíthatjuk, illetve közelíthetjük a képet, az 5-ös billentyű pedig a perspektivikus és a párhuzamos módok közötti váltásra szolgál.

A perspektivikus nézet természetesen szebb, és valósághűbb, de szerkesztéskor a párhuzamos sokkal átláthatóbb és egyszerűbb.

Nem árt megjegyezni, hogy az aktív panel mindig az, amelyik felett az egérmutató tartózkodik. Ha az egérmutatót a képernyő aljára visszük, akkor a billentyűzeten lenyomott gombok is az alsó panelre lesznek hatással. Ne csodálkozzunk tehát, ha ráközelítéskor (zoom) a gombok elkezdenek megnőni, vagy összemennek. Nem a ***Blender*** a hibás, csak az egérmutató van rossz helyen.

A másik módszer, amivel tetszőleges irányból szemlélhetjük a teret, az egér használata. Ha van görgőnk, akkor azzal is nagyíthatunk, illetve a középső gomb nyomva tartásával forgathatjuk a képet. Ha előtte a ***Shift*** gombot is lenyomjuk, akkor az egész képet egyben mozgathatjuk. Ha esetleg eltévednénk, és nem vagyunk benne biztosak, hogy honnan is látjuk az – amúgy minden oldalról egyforma – kockánkat, nyugodtan használjuk a 7, a 3 vagy az 1 gombok valamelyikét. Most, hogy remélhetőleg már minden oldalát megnéztük az unalmas kockánknak, jó lenne, ha tudnánk vele kezdeni valamit. Ehhez először meg kell tanulnunk kijelölni az objektumokat. Ha csak egy objektumot szeretnénk kijelölni, a jobb egérgombbal könnyedén megtehetjük. A kijelölt objektumok széle mindig rózsaszín. Egyszerre többet is szelektálhatunk, ha kattintás közben nyomva tartjuk a ***Shift*** billentyűt. A kijelölésből a ***Shift+Alt+jobbklikk*** kombinációval vehetünk ki egy objektumot. Megfigyelhető, hogy az utoljára kijelölt objektum mindig kicsit más színű, mint a többi, és a neve meg is jelenik a 3D view bal alsó sarkában. Ez azért van, mert bizonyos műveletek csak egy objektumon hajtódnak végre, vagy szükség lehet egy elsődleges objektumra. Ilyen lesz majd például a csoportosítás, ahol a modelljeinket hierarchiába rendezhetjük.

Gyík és Maszk – Mészáros (Henti) Balázs

Az *„A”* billentyűvel kijelölhetjük az összes objektumot, illetve le vehetjük róluk a kijelölést. Csoportos kijelölésre is van lehetőség. Ha például nem akarunk kézzel végigklikkelgetni 200, a képen lévő apró tárgyat, amik amúgy egymás mellett vannak, használhatjuk a ***Border Select*** opciót.

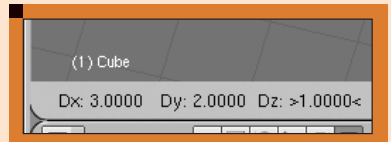
Ha megnyomjuk a *„B”* billentyűt majd a bal egér-


Azonosulás – Mészáros (Henti) Balázs

gombot használva egy téglalapot rajzolunk, az összes – a téglalapon belül lévő tárgy – kijelöltté válik. Ha ugyanezt a jobb egérgombbal próbáljuk, pont ellenkező hatást érünk el, levesszük a kijelölést.
Tehát jelöljük ki a kockánkat és nézzük elsősorban mit is lehet kezdeni vele. Háromféle transzformáció létezik (***rotate***, ***grab/move*** és ***scale***) amiket elég gyakran fogunk használni. Mindhárom transzformációt többféleképpen indíthatjuk, Menüből, billentyűzetről, egérrel, illetve az ***Objektum Tranform Properties*** ablakában kézzel beállíthatjuk. A három művelet:

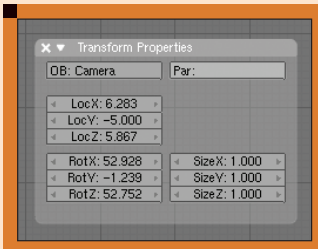
Grab/Move – A kijelölt tárgy(ak)at tudjuk vele mozgatni
Rotate – Egy megadott referenciapont körül tudunk forgatni
Scale – A kijelölt objektumokat tudjuk vele „skálázni”, vagyis méretarányosan növelni vagy csökkenteni.

A legelső bűvészmutatványunk az lesz, hogy kockánkat téglává varázsoljuk. Jelöljük is ki, majd nyomjuk meg az ***S (Scale)*** betűt. Ha most elkezdjük mozgatni az egeret láthatjuk, hogy kockánk méretarányosan változik, a ***3D view*** bal alsó sarkában pedig három tizedesjegy pontossággal azt is nyomon követhetjük, hogy hányszorosa lesz az eredetinek.
Ha úgy gondoljuk hogy elég nagyra/kicsire sikerült, megnyomhatjuk a bal egér gombot véglegesítve a transzformációt, vagy a jobb egérgombot, amitől vissza kapjuk az előző kockánkat, mintha semmi sem változott volna. Lehetőség van arra, hogy a három méret (X, Y, Z) közül csak egyiket változtassuk meg, tehát a kockánk hosszabb legyen, mint amilyen széles. Ettől persze elveszti „kockaságát”, de nekünk pont ez a célunk. Váltsunk tehát ***Scale*** módba (***S billentyű***), majd rögtön ez után nyomjuk meg az ***X billentyűt*** is. Láthatjuk, hogy most kockánknak csak az ***X*** koordinátája változik. Ha közben a ***CTRL*** gombot nyomva tartjuk, az alapméretezett négy tizedesjegy helyett most 1 tizedesjegy pontossággal állíthatjuk a méretet. Növeljük meg tehát a szélességét pont a kétszeresére, hosszát pedig 4szeresére és kész is a téglánk, most már csak helyére kell igazítani.
Ehhez jelöljük ki, majd nyomjuk meg a ***G billentyűt (Grab/Move)***. Most a téglánkat az éppen aktuális nézőpont síkjában szabadon mozgathatjuk. Itt is lehetőség van az ***X***, ***Y*** vagy ***Z*** billentyűk használatára, ha csak adott irányban akarunk elmozdulást.
A ***CTRL*** billentyű nyomva tartásával tizedesjegyek nélkül, a ***SHIFT+CTRL*** billentyűkombinációval pedig 1 tizedesjegy pontossággal mozoghatunk. Ha billentyűzetről szeretnénk megadni az elmozdulást, a ***G*** betű lenyomása után el is kezdhetjük beütni az X irányú elmozdulás értékét, TAB gomb után pedig az Y majd a Z koordinátát. A bal alsó sarokban nyomon követhetjük, hogy épp melyik koordinátát visszük be. Természetesen a ***Backspace*** is működik. (*7. ábra*)

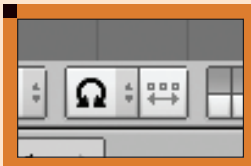


7. ábra Az elmozdulás értékét billentyűzetről is megadhatjuk

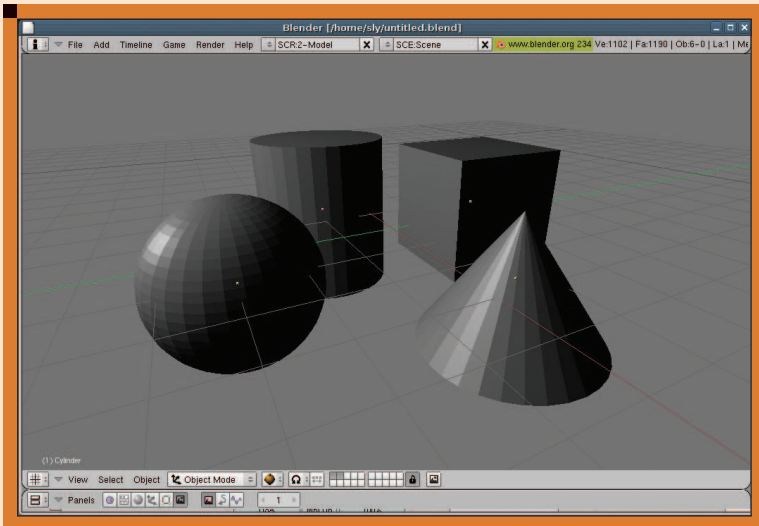

Jó étvágyat – Mészáros (Henti) Balázs



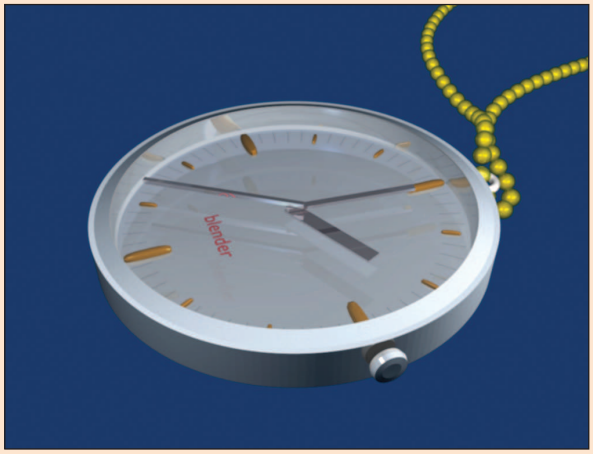
8. ábra Egy objektum tulajdonságai



9. ábra A referenciapontért felelős gombok



10. ábra Néhány „primitív”



Ha a nekünk megfelelő helyre mozgattuk a téglánkat, akkor akár el is forgathatjuk az ***R (Rotate)*** billentyűvel. A lehetőségek itt is ugyanazok: forgathatunk csak X, Y vagy Z tengely körül, használhatjuk a ***CTRL*** és/vagy ***SHIFT*** billentyűket a pontosabb eredmény érdekében, és billentyűzettel közvetlenül is megadhatjuk az elfordulás mértékét.
Ezek azonban csak relatív transzformációk, mindig az adott elhelyezkedéshez képest méreteznek át, forgatnak, vagy mozdítanak el. Ha szeretnénk megtudni, hogy hol is van pontosan a téglánk az origóhoz képest, kijelölés után nyomjuk meg az ***N gombot***. A megjelenő panelen az összes tulajdonságot állíthatjuk a globális koordináta rendszerhez képest, illetve az objektum nevét és ***szülőjét (Parent)*** is megváltoztathatjuk.

A gombok működése kicsi odafigyelést igényel: jobb oldalukra kattintva növelhetjük az értéket, bal oldallal csökkenthetjük, illetve középre kattintva átírhatjuk. Szorosan ide tartozik még a ***3D View*** fejlécén lévő két gomb *(9. ábra)*. A bal oldalival a referencia pontot tudjuk beállítani (mi körül forgatunk, melyik ponthoz képest nagyítunk/csökkentünk, stb), míg a jobb oldalival beállíthatjuk, hogy az egész objektumon, vagy csak annak középpontján akarjuk a transzformációt elvégezni. Referenciapontnak beállíthatjuk a ***3D Cursor***-t, az adott ***objektum középpontját (Median Point)***, vagy több kijelölt objektum esetén azt, hogy minden ***objektum saját középpontja körül forogjon el (Individual Object Centers)***.
Az ***Alt+G Alt+R*** és ***Alt+S*** billentyűkombinációkkal alapméretezett helyzetbe állíthatjuk az objektumunkat. Az ***Alt+G*** visszahelyezi a koordinátarendszer középpontjába, az ***Alt+R*** eredeti irányba fordítja, míg az ***Alt+S***-el az eredeti méretet kapjuk vissza. Használjátok egészséggel.

Hogy legyen min gyakorolni a transzformálást, meg kell tanulnunk a már meglévő téglánk mellé más primitív testeket is létrehozni. Vigyük az egeret a 3D nézet fölé, nyomjuk le a szóközt, majd válasszuk ki az ***Add*** menüpontot, azon belül is a ***Mesh***-t. Az itt látható objektumok az úgynevezett primitívek. Létrehozhatunk kockát, hengert, kúpot, gömböt (nem is egyfélét), de van egy meglepetés is, aki a ***„Monkey”*** névre hallgat. Ha hozzáadunk egy ilyen primitívet, az a 3D kurzor helyén fog megjelenni. Ezek után ugyanúgy transzformálhatjuk, mint a téglánkat, vagy ha meguntunk valamit, törölhetjük az X billentyűvel. Mindig figyeljünk oda, hogy mit jelöltünk ki, mert ***Object Mode***-ban nincs visszavonási lehetőség. Ajánlott a kényesebb műveletek után mindig menteni.
A következő számban tovább folytatjuk a téglánkkal való ügyetlenkedést, megtanulunk falat építeni belőle, majd elkezdjük csonkítani, illetve néhány mondatban az egyes panelek szerepét is megismerhetjük.
Addig is jó szórakozást!

Szalai András (sly87@freestart.hu)

Jelenleg középiskolába jár, ahol informatikát tanul. Jövőre érettségizik. Hobbija a programozás és a biztonságtechnika, és a továbbtanulási szándékai is ilyen irányúak.

iPod használata Linux alatt

Mit lehet megtenni egy Apple iPoddal és Linuxszal, amit az Apple iTunesával nem? Sok mindent.

A hordozható *MP3* lejátszók piaca robbanásszerűen bővült az utóbbi időkben, és az *Apple iPodját* sokan etalonnak tartják, minden lejátszót hozzá hasonlítva. Annak ellenére, hogy az *Apple* nem kínál linuxos változatot az *iTunes* zenekezelő alkalmazásból, azért az *iPod* a *Linux* felhasználók számára is kiváló *MP3* lejátszó marad.

Írásom születésekor egy 20 GB-os kapacitású, negyedik generációs *iPodot* használtam, „kattintókerékkel", illetve egy *iPod Shuffle*-t – ugyanakkor a szóba kerülő programnak az összes *iPoddal* együtt kell működnie. A *GTKPod* kiváló grafikus felületének használatához jobb, ha egy újabb *Linux* terjesztéssel rendelkezünk. Aki inkább a parancssorhoz vonzódik, az viszont gyakorlatilag bármilyen terjesztés alatt használni tudja a *GNUpodot*, amennyiben a *Perl* futtatására lehetősége van.

A *GTKPod* gyakorlatilag a nyílt világ válasza az *iTunesra*, segítségével, *iPodunk* és *Linuxunk* birtokában átmulathatjuk az éjszakát. A *GTKPod* egy csinos grafikus felület, mellyel *MP3*, *WAV* és *M4A* formátumú, *DRM* védelem nélküli *AAC* fájlokat másolhatunk át számítógépünkről *iPodunkra*. A fájlokat egyenként vagy könyvtáranként adhatjuk hozzá. Szükség szerint lejátszási listákat hozhatunk létre és szerkeszthetünk, normalizálhatjuk egyszerre akár több zeneszám hangerejét, illetve módosíthatjuk az *iPodon* és a számítógépen található fájlok *ID3* címkéit. A *GTKPod* elég okos ahhoz is, hogy az *iPodra* ne töltse fel a már meglévő fájlokat.

Természetesen a *GTKPod* a dalok az *iPodról* való letöltésére is alkalmas, ha kezdenénk kifogyni a helyből. A *GTKPod* a *GTK2*-re épül. Az eredetileg a *The GIMP*-hez fejlesztett *GTK*-t immár széles körben, számos alkalmazásban használják, illetve a *GNOME* asztali környezet is erre az eszközkészletre épül. A *GTK* ugyan nagyszerű szoftver, ám összetett is, számtalan függőséggel, keresztfüggőséggel és egyéb elvárással, vagyis a program saját kezűleg végzett lefordítása nagyjából annyira ígérkezik könnyű vállalkozásnak, mint például belebújni egy ürgelyukba. Mindannyiunk szerencsére minden a *GTK* alapú alkalmazások, így a *GTKPod* használatához szükséges dolognak már eleve rajta kell lennie a számítógépünkön, feltéve, hogy viszonylag új *Linux* terjesztést használunk.

Aki már használja az *iPodját*, az valószínűleg már minden olyan összetevővel rendelkezik, amely a *Linux* alatti használathoz szükséges: először is maga az *iPod*, egy számítógép *Firewire* vagy *USB 2.0* csatolóval, valamint egy viszonylag új *Linux* terjesztés. Szükségünk lesz a *GTKPod* másolatára is (lásd az internetes forrásokat). A *GTKPod* webhelyén rengeteg leírást találunk, hivatkozásokat a forráskódra, valamint *GTKPod* csomagokat *SUSE*, *Debian*, *Gentoo*, *Mandrake* és egyéb népszerű *Linux* terjesztésekhez. Az *rpm.pbone.net* vagy az *rpmfind.net* webhelyen keresve további *RPM* csomagokat is találhatunk.

Mivel az *iPod* gyakorlatilag egy csinos kis eltávolítható merevlemez, ha a *GTKPoddal*, *Linux* alatt akarjuk használni, akkor a rendszermagnak támogatnia kell a csinos kis eltávolítható merevlemezeket. Ez a fajta támogatás már a 2.4-es rendszermagban is megvolt, a 2.6-os változatban pedig tovább finomítottak rajta. A jelenlegi terjesztések szinte kivétel nélkül a 2.6-os rendszermagot használják, és az eltávolítható merevlemezek támogatása általában bele van fordítva a rendszermagba, esetleg külön betölthető rendszermagmodul formájában

Új iPodra FAT32

Ha az ***iPodunkat*** még csak most emeljük ki a dobozából, akkor valószínűleg módosítanunk kell a beállításain, mielőtt Linux alatt használni kezdenénk. Néhány beállítást át kell pöccinteni, egyszerűnek hangzik, nemde? A valóságban újra kell formázni az egészet. Rémisztő? Semmi ok az aggodalomra.

A legtöbb új ***iPodot HFS+*** fájlrendszerrel (ez a ***Mac OS*** fájlrendszere) megformázott merevlemezzel szállítják. Ha az ***iPodot*** gyári állapotában akarnánk ***Linux*** alatt használni, akkor a rendszermagnak ***HFS+*** támogatással is rendelkeznie kellene, valamint a ***Mac*** stílusú lemezrészeket is ismernie kellene. A legtöbb korszerű terjesztés mindezek önműködő támogatására nem képes, és ilyen jellegű szolgáltatásai csak újrafordítás után válnak elérhetőkké. Többen jelezték, hogy a ***HFS+*** rendszermagbeli támogatása jelenleg is hibás. Ha tehát csupán új játékszerünk kedvéért nem nagyon szeretnénk újrafordítani a rendszermagot, akkor folyamodjunk az egyszerűbb megoldáshoz.

Mivel a ***Microsoft Windows*** legtöbb változata erősen húzza az orrát, ha ***Mac*** fájlrendszert kell írnia vagy olvasnia, ***Windowsra*** végzett telepítéskor az ***iTunes*** program első teendője az ***iPod*** beállításainak megváltoztatása. Igen, az ***iTunes*** következetesen beállításokról beszél, ám a valóságban újraformázza az ***iPod*** belső merevlemezét, méghozzá windowsos ***FAT32*** fájlrendszerrel. Ha tehát most vettük ki a dobozból az ***iPodot***, és hozzá tudunk férni egy windowsos géphez, akkor az ***iTunes*** program telepítőjével elvégezhetjük az ***iPod*** formázását. Az ***iPod Shuffle*** már eleve ***FAT32*** fájlrendszerrel formázva kerül ki a gyárból, vagyis gyári állapotában is használni tudjuk.

Az ***iPod*** újraformázását a windowsos program ***Restore iPod*** (***iPod*** helyreállítása) szolgáltatásával is elvégezhetjük, ekkor azonban ügyeljünk arra, hogy a művelet teljesen legyalulja az ***iPod*** tartalmát, vagyis minden dalt és fájlt elveszítünk. Aki az ***iPodját Mac*** géppel használta, és most linuxossal szeretné, de az újraformázást el akarja kerülni, az az internetes források között talál olyan hivatkozásokat, amelyek pontosan ilyen áttérést végrehajtó ***Mac***-használók tanácsait foglalják össze. Aki sem ***Windowst***, sem ***Mac OS***-t nem akar használni, illetve aki nem tud windowsos géphez hozzáférni, az szintén nézzen körül a források között, mert az ***iPod*** kizárólag ***Linux*** használatával végzett újraformázásához is talál útmutatást. A dolog természetesen megoldható, de a ***HFS+*** és a ***Mac***-lemezrészek támogatását igényli a rendszermag részéről, illetve a lecsupaszított ***iPod*** alapszintű könyvtárszerkezetének létrehozását a ***GNUPoddal*** kell elvégezni.

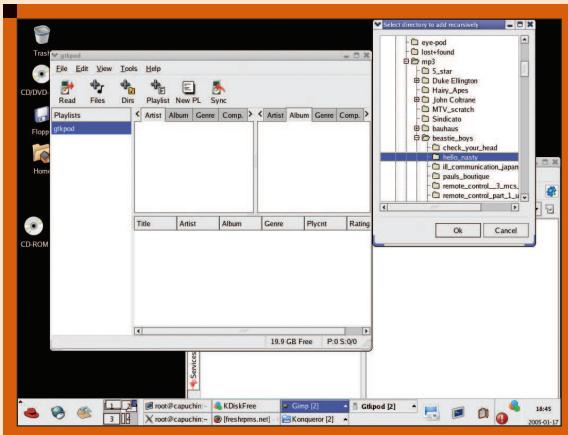
iPod + Linux = iPodLinux

Az ***iPodLinux Project*** keretein belül folyó munka eredményeként ***iPodra*** is telepíthetünk ***Linuxot***. Az ***iPodLinux*** az ***uClinux*** rendszermag átültetett, az ***iPod*** hardverére szabott változatára épül, grafikus felületét a ***Podzilla*** biztosítja. Az ***iPodLinux*** lehetővé teszi, hogy rendszerindításkor válasszunk, a ***Podzillát*** vagy a normál ***iPod*** operációs rendszer kívánjuk elindítani, ha tehát szeretnénk valami újat és mást kipróbálni, ismerkedjünk meg vele.

áll rendelkezésre. Röviden és tömören: elvileg semmire nincs szükség a rendszeren, hogy képes legyen látni az ***iPodot***.
Linuxos gépünkhöz ***Firewire*** vagy USB 2.0 kapcsolaton keresztül csatlakoztathatjuk az ***iPodot***. Az ***USB 1.1*** összeköttetések is működnek, de mert jóval lassabbak, nem érdemes ilyet használni. Az ***USB 1.1*** egy másik hátránya, hogy az ***iPod*** nem tud feltöltődni, miközben a számítógéphez van csatlakoztatva.
A ***GTKPod*** telepítésével nem lehet gondunk, akár forrásból fordítjuk, akár előre lefordított, bináris csomagot teszünk fel. A ***GTKPodnak*** szüksége van a ***libid3tag*** könyvtárra, a ***libmp4v2*** csomagot pedig akkor kell rendelkezésére bocsátanunk, ha ***AAC*** fájlokat is használni szeretnénk (lásd a forrásokat). A ***GTKPod*** forráskódja a megszokott

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

parancsokkal fordítható le. Érdemes azonban előre lefordított, bináris csomagot választani, hiszen ennek telepítése könnyebb, és meglévő asztali környezetünkkel szorosabban egybe tudjuk építeni.
Ha túlestünk az ***iPod FAT32***-re formázáson és a ***GTKPod*** telepítésén, csatlakoztassuk a mellékelt ***Firewire*** vagy ***USB*** kábelt a számítógéphez, majd a másik végét az ***iPodhoz***. Ha a ***Linux*** látja az ***iPodot***, akkor az ***iPod*** kijelzőjén rövid időre megjelenik a ***„Do Not Disconnect" (Ne szakítsa meg a kapcsolatot)*** felirat, az ***iPod Shuffle*** állapotjelzője pedig narancs színnel villogni kezd. Ha az ***iPodunk*** nem így reagál, próbáljuk meg lehúzni és újra csatlakoztatni. Én a saját 20 GB-os ***iPodommal Firewire***-t használok, nem ***USB***-t, ugyanis az interneten fellelhető források nagy része a ***Firewire*** alkalmazását tárgyalja. Ha csak egy aktív ***Firewire*** eszközzel rendelkezünk, akkor az ***iPod*** leválasztása is egyszerűbb egy kicsit. Az iPod Shuffle ugyanolyan könnyen és megbízhatóan csatlakozik, mint az ***USB***-kulcsok. Akinek a ***Linuxa*** nem látja az ***iPodot***, az írásom ***Hibaelhárítás*** című részében talál tanácsokat. Ha lemezes, már eleve ***FAT32*** fájlrendszerrel formázott ***iPodot*** használunk (lásd a széljegyzetet), akkor a meghaj-



1. ábra Az GTKPod a zenéket a szokásos jellemzők alapján rendezi, és lejátszási listák létrehozását is lehetővé teszi

tón két lemezrész található. Az egyik körülbelül 40 MB-os, ezen van az ***iPod*** belső operációs rendszere, a másik, nagyobb méretű lemezrészen pedig a zenék tárolása történik. Ha ***iPod Shuffle***-t használunk, akkor csak egy lemezrészünk van, és ezt gyárilag is ***FAT32*** fájlrendszerrel formázták meg. Mivel minden számítógép különbözik egy kicsit a többitől, javaslom a ***/var/log/messages*** fájl, illetve az adott terjesztésen a neki megfelelő fájl átnézését, ebből kiderül, hogy mi történik a színfalak mögött, amikor először csatlakoztatjuk az ***iPodunkat***. A naplófájl üzenetei szerint a rendszernek fel kell ismernie az ***iPodot***, majd hozzá kell rendelnie a következő ***SCSI*** meghajtójelet, ami, ha csak ***IDE*** meghajtókat használunk, a ***/dev/sda***. Egyes rendszerek az ***fstab-sync*** segédprogrammal önműködően módosítják a ***/etc/fstab*** fájlt, és beillesztenek egy befűzési pontot az ***iPod*** számára, mások viszont elvárják, hogy a ***/etc/fstab*** fájlban már eleve legyen egy megfelelő bejegyzés. Ha belenézünk a ***/var/log/messages*** fájlba, és nem látunk valami hasonlót:

```
fstab-sync[4284]: added mount
➥point /mnt/ipod for /dev/sda2
```

akkor adjuk hozzá kézzel a következő sort a ***/etc/fstab*** fájlhoz:

```
/dev/sda2 /mnt/ipod vfat
➥rw,user 0 0
```

Ne feledjük, hogy a merevlemezes ***iPodok*** esetében csak a második, a zenéket tároló lemezrészt (sda2) kell befűznünk, a kisebbet – amin az operációs rendszer található (sda1) – nem. Az ***iPod Shuffle*** esetében, mely csak egy lemezrésszel rendelkezik, a következő bejegyzést kell alkalmaznunk:

```
/dev/sda1 /mnt/ipod vfat
➥rw,user 0 0
```

Ha még nem tettük meg, az `mkdir /mnt/ipod` parancsot rootként kiadva hozzuk létre a befűzési pontot. Alapesetben a ***GTKPod*** megkeresi a ***/mnt/ipod*** alá befűzendő ***iPodot***, de ezen módosíthatunk is. Az ***iPod*** befűzésére a legegyszerűbb módszer az, ha a feladatot a GTKPodra bízzuk. Ehhez indítsuk el a ***GTKPodot***, majd válasszuk az ***Edit (Szerkesztés)*** és az ***Edit Preferences (Beállítások szerkesztése)*** parancsot. Az ***Input/Output (Kimenet/bemenet)*** lapon jelöljük be a ***Handle mounting/umounting of iPod drive (iPod meghajtó befűzésének/leválasztásának kezelése)*** jelölőnégyzetet. Indítsuk újra a ***GTKPodot***, és az ***iPodunk*** befűzése önműködően megtörténik.

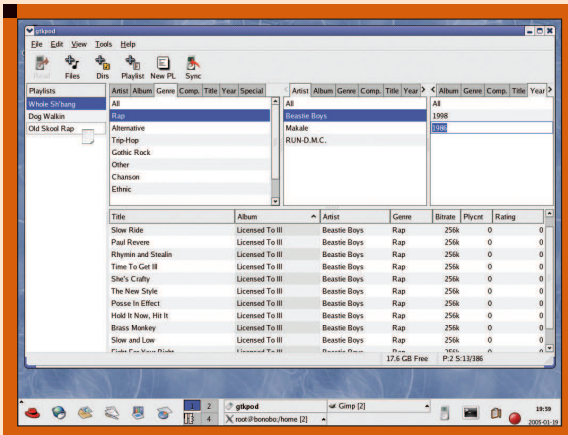
A ***GTKPod*** számos beállítást és szolgáltatást támogat, megismerésének legjobb módja, ha átmásoltatunk vele néhány zeneszámot az ***iPodra***. Bízzunk benne, hogy számítógépünkön eleve jó néhány ***MP3*** fájl található. A ***GTKPod*** csak az ***iPodon*** lévő fájlok kezelésére alkalmas, magukat az ***MP3*** fájlokat valamilyen másik programmal kell elkészítenünk, mint például a ***Grip***. Az ***MP3***-ak meghallgatását a ***GTKPod*** szintén csak külső programmal teszi lehetővé, mint például az ***XMMS***; kedvenc ***MP3*** lejátszónkat a ***GTKPod*** beállításai között adhatjuk meg.

Ha beszereztünk néhány áttölthető ***MP3***-at, akkor az ***Add Files (Fájlok hozzáadása)*** gombbal egyes fájlokat tudunk hozzáadni, az ***Add Dirs (Könyvtárak hozzáadása)*** gombbal pedig egyszerre egész könyvtárakat. Az *1. ábrán* követhető, hogy a ***Beastie Boys Hello Nasty*** albumát hogyan adtam hozzá a frissen formázott ***iPodomhoz***. Amikor egy könyvtár hozzáadásakor rákattintunk az ***OK*** gombra, a ***GTKPod*** végigmegy a fájlokon, hozzáadja őket a helyi ***iTunes*** adatbázishoz (ez alapesetben a ~/.gtkpod/ könyvtárban található), majd az alsó panelen előadó, cím, album vagy műfaj szerint rendezve kilistázza őket. A folyamat következő lépéseként kattintsunk a ***Sync (Szinkronizálás)*** gombra. A fájlok tényleges átvitele a számítógépről az ***iPodra*** csak ekkor történik meg, illetve a számítógép helyi ***iTunes*** adatbázisa is szinkronizálásra kerül az ***iPodon*** találhatóval. A másolás a kiválasztott zene mennyiségétől függően egy-két percig tart. Miután a fájlok másolása sikeresen befejeződött, a ***GTKPod*** ablakának bal alsó sarkában az ***„iPod Database Saved" (iPod adatbázis elmentve)*** felirat jelenik meg.

Ekkor azonban még ne távolítsuk el az ***iPodot***. Ha az ***iPodon*** a ***„Do not disconnect" (Ne szakítsa meg a kapcsolatot)*** felirat látható, illetve a ***Shuffle*** állapotjelző fénye továbbra is narancs színnel villog, akkor hagyjuk a készülékeket összekapcsolva. Ne feledjük, az ***iPodot*** mint eltávolítható merevlemezt fűztük be. Egy még pörgő merevlemezt sem húznánk csak úgy le a gépről, igaz? Ügyeljünk tehát arra, hogy megtörténjen az ***iPod*** leválasztása. Ha a ***GTKPod*** alatt önműködő befűzést használunk, akkor egyszerűen lépjünk ki a ***GTKPodból***, és megtörténik az ***iPod*** leválasztása.

Lehetséges, hogy az ***iPod*** továbbra is a ***„Do not disconnect"*** figyelmeztetést jeleníti meg. Az ***iPodot*** csak akkor szabad lehúzni, ha az üzenet eltűnt, ezt pedig az eltávolítható merevlemezek kezeléséért felelős rendszermagmodul eltávolításával tudjuk elérni. Ha ***Firewire*** kapcsolatot használunk,

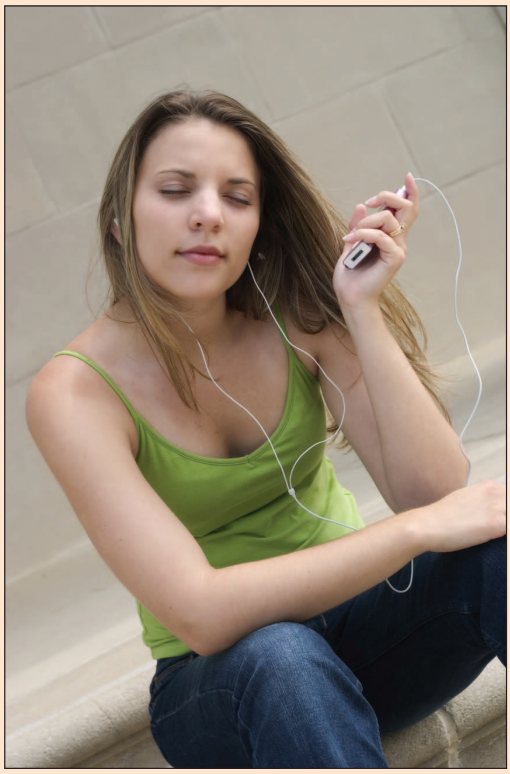
akkor az *sbp2* modult kell eltávolítanunk, ezt a root terminálján a `modprobe -r sbp2` parancsot kiadva tehetjük meg. Ha a *„Do not disconnect”* üzenet eltűnt, és láthatóvá válik az *iPod* normál menüje, akkor végre lehúzhatjuk a számítógépről.
A *GTKPod* indításakor és bezárásakor képes parancsfájlok önműködő elindítására is. A *~/.gtkpod/gtkpod.in* és a *~/.gtkpod/gtkpod.out* – illetve a */etc/gtkpod.in* és a */etc/gtkpod.out*, ha a fájlok nem a kezdőkönyvtárunkba kerülnek – fájl kiolvasása a *GTKPod* minden elindításakor és bezárásakor megtörténik. Indításkor a parancsfájl futtatása az *iPod* befűzése előtt történik meg. Ha tehát az *iPod* használatához be kell töltenünk egy rendszermagmodult, esetleg egyéb rendszerműveletet kell elvégeznünk, akkor ebből a parancsfájlból megtehetjük. Hasonlóan a *gtkpod.out* fájl révén, a megfelelő parancsfájlokkal egyetlen pillanat alatt elvégezhető az *iPod* eltávolítása.
Ha az eltávolítás megtörtént, akkor az összes áttöltött zeneszámot le tudjuk játszani. Akinél a dolgok nem a tervezett módon működnek, az fussa át a cikk Hibaelhárítás című részét. Ha sikerrel jártunk, próbáljunk további zenéket hozzáadni, eközben fedezzük fel a *GTKPod* néhány a zeneszámok kezelésével kapcsolatos szolgáltatását.
Az *iPodon* található fájlok kezeléséhez kattintsunk a *GTKPod Read (Olvasás)* gombjára, ekkor a program kiolvassa az *iPodon* lévő *iTunes* adatbázis tartalmát. A programnak meg kell jelenítenie az imént hozzáadott dalokat. A bal szélső panel mutatja, hogy melyik lejátszási lista van kiválasztva. A lejátszási lista pontosan az, aminek hangzik: zeneszámok egy csoportja, a zeneszámokat egymást követően tudunk lejátszani. A lapokra osztott ablakban a zenék az *iPod iTunes* adatbázisában található adatok szerint szerepelnek, és a kiválasztott lapok alapján jelennek meg. Alapesetben csak két panel van, de a beállítások módosításával továbbiakat is hozzáadhatunk. Én egy további panel hozzáadásával kaptam a három panelt tartalmazó elrendezést, így pontosabban szét tudom válogatni a zenéimet.
Lejátszási listákat létrehozva saját magunk lemezlovasai lehetünk, és az általunk kedvelt mixeket akkor



2. ábra A lejátszási listákat könnyedén, kattint és húz módszerrel állíthatjuk össze

játszhatjuk le, amikor csak akarjuk. Tudom, az ízlések és a pofonok sokfélék lehetnek, de nekem semmi sem dobja fel jobban a napomat, mint egy kis klasszikus, keleti parti rap. Készítek tehát egy lejátszási listát az irodában töltött sötét napokra, amikor egy csésze kávé nem elég ahhoz, hogy felpörögjek. Rákattintok a *New PL (Új lejátszási lista)* elemre, létrehozom az *Old Skool Rap (Régi stílusú rap)* lejátszási listát, majd a *GTKPodban* végzek egy rendezést, és kiválasztom a megfelelő zeneszámokat (*2. ábra*). A külön panel segítségével műfaj, előadó és évjárat alapján tudok rendezni. *Beastie Boyst* akarok hallgatni, de csak a régi számokat szeretem, vagyis rákattintok 1986-ra, majd behúzom az új lejátszási listámba, amibe ezzel 13 fájl másolódik be. Ez után továbblépek a *RUN-DMC*-re, és elvégzem ugyanezt. Közben rájövök, hogy még le kellene másolnom *CD*-lemezről néhány számot, vagyis munkába állítom a *Gripet*.
Aki már használ más alkalmazást a lejátszási listák összeállítására, az a meglévő listákat is használhatja a *GTKPod* alatt. A *GTKPodnak* nem okozhat gondot a meglévő *.m3u* vagy *.pls* fájlok lejátszási listaként való alkalmazása. Csak kattintsunk a *New PL* gombra, adjunk nevet a lejátszási listának, kattintsunk rá az *Add File* gombra, majd keressük mg a kívánt lejátszási listát. A *GTKPod* a meglévő lejátszási listák *m3u* formátumú kimentésére is képes.
A *GTKPod* szolgáltatásai közül én a leginkább azt kedvelem, hogy alkalmas az *MP3* fájlok *ID3* címkéinek szerkesztésére, legyenek azok akár a számítógépen, akár az *iPodon*. Az *ID3* címke az *MP3* fájl metaadatokat – előadó neve, album címe, dal címe, kiadás éve – tartalmazó része. Magát az *MP3* fájlt bárhogyan elnevezhetjük, a *GTKPod* és az *iPod* kizárólag az *ID3* címkére támaszkodik. Más részről, ha egy *MP3* fájlban valamiért nincs *ID3* címke, akkor a különféle listákban nem jelenik meg, a *GTKPod* alatt csak üres fejlécű fájlként látható, az *iPodon* pedig az összes fájlt meg kell jeleníteni, hogy elérhetővé váljék. A problémát úgy háríthatjuk el, hogy *GTKPod* alatt rákattintunk a megfelelő részre, és begépeljük a kívánt adatokat.
Ha az alsó ablakpanelen adott dalra kattintunk, akkor csak az adott fájl *ID3* címkéjét tudjuk szerkeszteni.
Ha dalok egy csoportjában ugyanazt a mezőt szeretnénk szerkeszteni, akkor valamelyik rendezőablakban szerkesszük át a mezőt, a változások az alsó panel összes fájljára érvényesek lesznek. Több fájl *ID3* címkéjének egyidejű módosítására a *Multi-Edit* szolgáltatást is használhatjuk. A szolgáltatás kiegészítő jellegű, a beállítások között külön kell engedélyezni, de lehetővé teszi, hogy – a Windowsban is megszokott módon – a Shift gombbal több fájlt is kijelöljünk, majd a módosított mező, például az előadó tartalmát az összes dalra alkalmazzuk.
A *GTKPod* további nagyszerű szolgáltatása, ami az *Apple iTunes* alkalmazásából hiányzik is, az a dalok kimentésének lehetősége, vagyis visszamásolása az *iPodról* a számítógépre. Természetesen ezt úgy is megtehetjük, hogy egyszerűen befűzzük az *iPodot* mint


Linuxvilág
www.linuxvilag.hu



külső merevlemezt, majd addig keresgélünk rajta, amíg meg nem találjuk a keresett fájlokat, ám sokkal egyszerűbb a *GTKPod* segítségével rendezni a tartalmat, és ezt követően választani ki a másolni kívánt fájlokat. A művelethez a *File (Fájl)* menü *Copy Tracks from iPod (Zeneszámok másolása az iPodról)* parancsára lesz szükségünk. Az *Edit (Szerkesztés)* menü *Delete Completely From iPod (Teljes törlés az iPodról)* beállítása a neve alapján várható módon működik, alkalmazásával értékes megabájtokat szabadíthatunk fel az *iPodon*, ami jól jöhet az éppen a kedvenceink közé tartozó dalok tárolásához.

A *GTKPod* feltűnően hiányzó szolgáltatása az *iTunes* zeneboltból vásárolt zeneszámok kezelése. Ezek az *iTMS* zeneszámok *AAC* formátumba vannak tömörítve, majd *DRM* technológiával „csomagolják” őket, ami meghatározza, hogy az egyes számokkal mit tehetünk meg. Ha nagymennyiségű zenét vásároltunk össze az *iTunesról*, és ezeket a *GTKPoddal* szeretnénk kezelni, akkor két választási lehetőségünk van. Az első, hogy az *iTunes* segítségével *CD*-lemezre írjuk, majd például a *Grippel* visszamásoljuk őket. Ez a megoldás ugyan működik, de amíg nem teljes albumokkal dolgozunk, addig a *CDDB* adatbázis nem tud mit kezdeni a lemezzel, vagyis az összes *ID3* címkét kézzel kell beírnunk. A második lehetőség, hogy például a *Hymn* segítségével az összes *DRM*-es rondaságot letisztítjuk a megvásárolt zeneszámokról. Mindenkit figyelmeztetnék, hogy a *Hymn* kijátssza a *DRM*-et, és használata jogsértő azokban az országokban, ahol a gyártókhoz való kötődést engedélyező törvény létezik; ilyen például az *Amerikai Egyesült Államok Digital Millennium Copyright Actje*.

Aki nem kedveli a grafikus felületeket, inkább a héjak egyszerű eleganciája vonzza, az ismerkedjen meg a *GNUpoddal*; mely az *iPodon* található zenék kezelését segítő Perl fájlok gyűjteménye. A *GNUpod* eszközeivel mindent el lehet végezni, kezdve az *iPod* a zeneszámok tárolására szolgáló könyvtárszerkezetének létrehozásától, a zenék hozzáadásán és törlésén keresztül egészen a lejátszási listák kezeléséig. Mindezt a parancssorból lehet véghez vinni, a Perl parancsfájloknak különféle átadott értékeket adva, mint például:

```
gnupod_addsong.pl -m /mnt/ipod
➥/tunes/rappers_delight.mp3
```

A *GNUpod* telepítése néhány másodperc alatt elvégezhető. A közösség tagjai által készített, az interneten elérhető leírások minden részletre kiterjednek, a lépésről lépésre haladó útmutatásokra támaszkodva a *Firewire* üzembe helyezésétől egészen az *iPod* újraformázásáig mindent megcsinálhatunk. Írásom születésekor a *GNUpod* az összes *iPod* támogatására képes volt, ide értve a *Shuffle*-t is. A *GTKPod* jelenlegi változata (0.87) a *Shuffle*-t nem támogatja, de lehetséges, hogy a megjelenés időpontjára már továbbfejlesztik.

Hibaelhárítás

Ha problémába ütközünk, semmi esetre se rémüljünk meg! Nálam is voltak kisebb fennakadások, mire minden megfelelően működni kezdett. Először is, nem árt tisztában lenni azzal, hogy az illesztőprogramok egy része nem tökéletes. Az *iPod Firewire* kapcsolat feletti használatakor az *sbp2* és az *ohci1394* modulra van szükség. A tesztrendszereim egyike *Red Hat Fedora Core 3* alapú volt, melyen az *up2date* segítségével letöltött és telepített *2.6.10-1.741_FC3* rendszermag futott. Megjegyzem, ennek a terjesztésnek a gyári rendszermagja ismerten hibás, ami miatt az iPod *USB* kapcsolaton keresztüli használata rémálom. A dolgok nagy része zökkenőmentesen működött, bár az *sbp2* modul összeomlott, amikor rootként megpróbáltam eltávolítani a `modprobe -r sbp2` paranccsal. Egyszer olyan is volt, hogy újra kellett töltenem az *ohci1394* illesztőprogramot, mert az *iPod* nem látszott. Természetesen mindenki más és más utat jár be, függően a rendszermag változatától és a gépében található *Firewire* kártya lapkakészletétől.

Akinél az *iPod* lehúzása, újracsatlakoztatása vagy a modulok eltávolítása és ismételt betöltése után sem látható az *iPod*, az se essen kétségbe. Nálam is volt ilyen, az egyetlen járható út, amit találtam, a gép újraindítása volt úgy, hogy az *iPod* csatlakoztatva volt hozzá. Nem mondanám, hogy elegáns megoldás, de hatékony. Mindenkinek javaslom, hogy olvassa át a terjesztéséhez tartozó leírást, annak alapján biztosan könnyebb lesz a felbukkanó hibák elhárítása.

Linux Journal 2005. július, 135. szám

A cikkhez tartozó források elérhetősége:
➲ *www.linuxjournal.com/article/8210*

Bert Hayes a 2.0-s rendszermag homályba vesző megjelenése óta Linux felhasználó és rendszergazda. RHCE, a Snort for Dummies társszerzője. Hobbija a kerékpározás és egy léghűtéses VW busz restaurálása.

3D ábrázolás a PoVRay segítségével (1. rész)

A számítógépek megjelenése előtt is már voltak törekvések a valós tárgyak ábrázolására két dimenziós felületen, gondoljunk csak az ókori és középkori freskókra és festményekre. A mérnökök is szerették az általuk használt műszaki rajzolásban különféle eljárásokat használni, amelyekkel már a tervezőasztalon elképzelhető lett egy-egy munkadarab. A két dimenziós felületeken leképezett három dimenziós világnak nagy hagyománya van, bármennyire is törnek előre a valódi 3D megjelenítők.

Alapelvek

Minden matematikai és technikai elvet mellőzve annyit lehet kijelenteni, hogy a 3D ábrázolásnak kettő elterjedt módszere van, amelyeket a különféle megjelenítő programok használnak: a ***poligon alapú ábrázolás*** és a ***sugárkövetés***.

A poligonok ábrázolása során a számítógép általában háromszögekből készíti el az ábrázolni kívánt test felületét, s a műszaki rajzok ábrázolásából szivárgott át más területekre is. Egy kockához hat négyszög kell, azaz 12 háromszög, így viszonylag kevés koordinátával leírható lesz a test. A síklapok térbeli elhelyezése nagyon kevés számítással elvégezhető. Egy gömb ábrázolásához viszont már sok száz-sok ezer háromszöget kell meghatározni, csak a sík felületek leírásához célszerű poligonokból összeállítani egy-egy testet. Kétségtelen, hogy a számítógépes játékok elterjedése a poligonok ábrázolása révén vált lehetővé, mivel ez a módszer kevesebb számítási erőforrást igényel, viszont az elkészült „világ" is szögletes és elnagyolt lesz. Ha növeljük a poligonok számát, a minőség és a szögletesség is lassan eltűnik, viszont – mivel felületekről van szó – négyzetesen növekszik a szükséges számításigény. Különféle trükkök szükségesek, hogy a fénybe helyezett tárgy árnyékot vessen (***shading***), vagy a felületén az egyenetlenségeket észrevehetően ábrázoljuk (***bump mapping***). Ha tükröződés is cél, akkor a poligonok ábrázolása már kevés lesz, kénytelenek leszünk némi sugárkövetést is alkalmazni.

A sugárkövetés szinte egy időben jelent meg a számítógépekkel, mivel pontos és összetett számolások ismételgetése szükséges egy kép összeállításához. A sugárkövetés valójában a létező fizikai világ modellezése, a fényforrásból induló különböző energiájú (színű) fotonok a tárgyakon áthaladnak, elnyelődnek, újra kibocsátódnak, szóródnak, visszaverődnek, módosulnak és végül egy nagyon kis részük a szemlélődő ember szeméig jutnak el. A sugárkövetés ezen okból fordítva működik a számítógépes valóságban, ugyanis pazarlás olyan sugarakat követni, amelyek nem jutnak el a megfigyelőig. Ennek megfelelően az egész eljárás vektorok kezeléséből áll, a nézőpontból (kamera) a képernyő felbontásának megfelelő számú „látósugár" indul meg a virtuális világ felé, ahol kölcsönhatásba lépnek a testek modelljeivel, s végül egy fényforrásba vagy a végtelenbe jutnak el. Nagy felbontás és sok test esetén többmilliárd visszaverődés és szóródás is lehetséges, illetve több fényforrásban is „véget érhet" egy-egy sugár élete. Megfelelő számítási kapacitás hiányában a sugárkövetés nem alkalmas valós idejű 3D mozgások ábrázolására, így a számítógépes játékok sem tudják kihasználni a tudását; a számítógéppel készített filmek és egyéb tévéjátékok viszont előszeretettel alkalmaznak sugárkövetést is, hiszen van elegendő idejük a használatára.

PoVRay – Persistence of Vision Raytracer

David K. Buck és ***Aaron A. Collins*** által készített ***DKBTrace 2.12*** program alapján készült el a ***PoVRay*** első verziója, amelyet szakemberek tucatjai fejlesztenek azóta is, s a forrása is hozzáférhető bárki számára.

A programnak saját speciális leíró nyelve van, amelyet ***SDL*** (***Scene Description Language***) rövidítéssel illetnek. Kismértékben hasonlít a C nyelvhez – minden bizonnyal a C nyelv volt a minta, de az a hasonlóság csak nagyon távoli rokonság.

A program maga csak parancssoros felülettel rendelkezik, bár van sok frontend hozzá, amelyekkel kényelmesen tudjuk használni (***Kpovray***, ***Blender***, stb).

Első feladatként készítsünk egy zöld színű gömböt, amelyet egy pontszerű fehér fényforrás világít meg megközelítőleg a kamera felől, s a kamera pont a gömbre figyel.

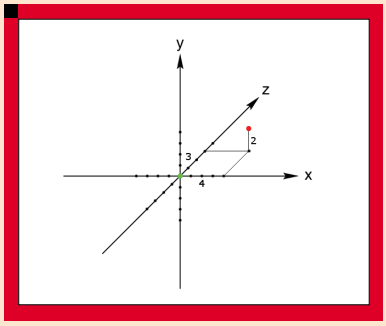
```
#include "colors.inc"

sphere{
 <0, 0, 0>, 2
 texture{
  pigment{
   color Green}}}
```

```
light_source{
 <2,4,3> color White}

camera{
 location <0,2,3>
 look_at <0,0,0>}
```

Nevezzük *pov01.pov* néven, s helyezzük bele egy tetszőleges könyvtárba, amelyet arra a célra fogunk szánni, hogy képeket és *PoVRay* „programokat” helyezünk el benne. Ezt a fájlt le kell „fordítanunk” ahhoz, hogy képet kapjunk kimenetként. Ez a fordítási művelet ***renderelés*** álnéven dolgozik, s a következő parancssorral tudjuk előcsalogatni:

```
povray +L/usr/local/share/
➥povray-3.6/include/ +H600
➥+W800 +Ipov01.pov +Opov01
```

Figyeljünk arra, hogy a `+L` paraméter után a *PoVRay* tényleges elérési útját adjuk meg, ez ugyanis minden disztribúcióban más lehet. A `+H` paraméter a kép magasságát, a `+W` a kép szélességét határozza meg, a `+I` utáni szöveg határozza meg a lefordítandó állomány elérési útját, s végül a `+O` a készítendő kép neve lesz. Alapértelmezésben *png* típusa lesz a képnek (régebben *tga* formátumot használtak a készítők), de ezt felül lehet bírálni, s a program szinte az összes gyakori képformátumot képes felhasználni. Ha a grafikus felületünk elérhető a program számára a futásának idejében, akkor egy ablakot kapunk, ahol követhetjük a kép készítésének menetét.

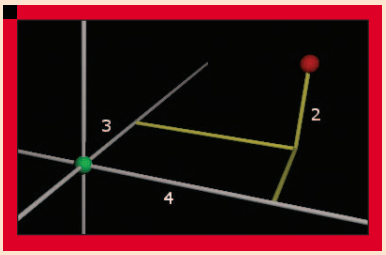


A 3D világ olyan koordináta rendszert kíván meg, amellyel meghatározhatunk pontokat a térben. Erre a célra a műszaki ábrázolás már kiötlött módszereket: egyszerűen a kettő dimenziós *x-y* koordináta rendszert kiegészítették egy harmadik – *z* jelű – tengellyel is, amely a megjelenítés síkjától tart tőlünk a végtelenbe. Egy pont megadása tehát három szám segítségével történik, mégpedig rendre az *x*, az *y* és a *z* koordinátával. Ha egy piros gömböt szeretnénk meghatározni a tér x=4, y=2 és z=3 pontjára, akkor ezt a *PoVRay* formátumában a

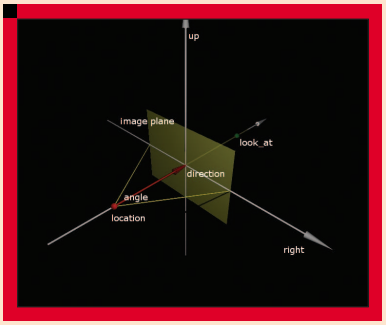
```
sphere{
 <4, 2, 3>, 0.2
 texture{
  pigment{
   color Red}}}
```

SDL részlettel tudjuk megoldani. Sokkal látványosabb lehet ez a *PoVRaY* által készített képet nézve amely egy példaprogramnak is tekinthető.

```
#include "colors.inc"

sphere{ // Zöld gömb az
        ➥origóban
 <0, 0, 0>, 0.2
 texture{
  pigment{
   color Green}}}

sphere{ // Piros gömb a
        ➥megadott pozícióban
 <4, 2, 3>, 0.2
 texture{
  pigment{
   color Red}}}

[...]

cylinder{ // A Z tengely
 <0,0,-10>, <0,0,10>, 0.05
 pigment{
  color White}}

[...]

cylinder{ // 3 egység hosszú
        ➥szakasz, Z
        ➥tengellyel párhuzamos
 <4,0,0>, <4,0,3>, 0.05
 pigment{
  color Yellow}}

camera{
 location <5,5,-10>
 look_at <0,0,0>}

light_source{
 <5,5,-5> color White}
```



A *pov02.pov* példa alapján – feltéve, ha erős mérnöki vizuális látással bírunk – azonnal „élni” látjuk a számokat. Ha pusztán az *SDL* fájlt szeretnénk használni, mindenképpen szükségünk lesz egyfajta belső látásra, amely biztosan kifejlődik a próbálkozások során. Nézzük tehát a virtuális világ alkotóelemeit, kezdjük a kamerával.
A *PoVRay* kamerája (*kamera.pov*) egyszerű felépítéssel bír, amely a látható képet is meghatározza. Bármerre is legyen a kamera (`location`), mindig van egy vektor (`direction`), amely meghatározza a képsík (`image plane`) középpontját. Ha megadunk nézőpontot (***`look_at`***), akkor a kamera átszámolja az irányvektorát, és a képsík közepén az a tárgy lesz, amelyet a nézőpont meghatároz. A képsík mérete alapesetben egységnyi, szélességét és magasságát közvetlenül a magasságvektor (`up`) és a jobbra mutató vektor (`right`) határozza meg, s ebből adódik a látószög (`angle`), amelyet megadunk közvetlenül is, ám ekkor módosulnak az egyéb vektorok is.



A program akkor is használ kamerát, ha erre külön nem utasítottuk. Az alapértelmezett kamera *SDL* leírója mindig felhasználásra kerül, s a saját kameránk csak felülírja pár alkotóelemét:

```
camera{
 perspective
 location <0,0,0>
```

```
 direction <0,0,1>
 right 1.33*x
 up y
 sky <0,1,0>
}
```

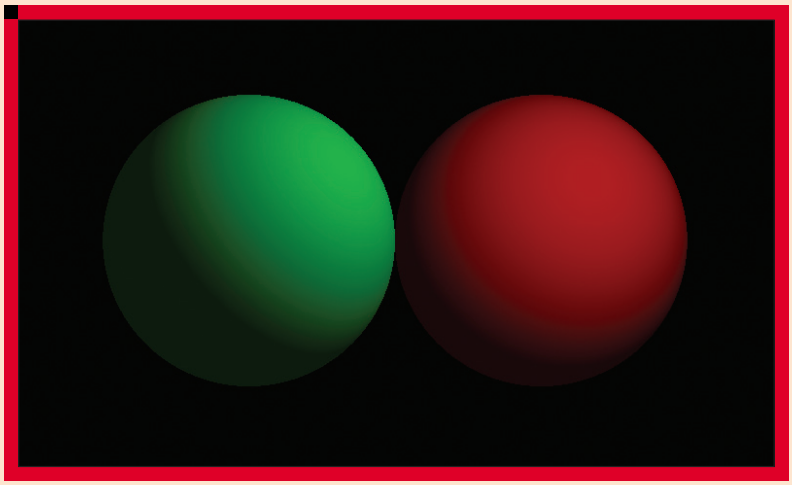
Az első kulcsszó jelenti, hogy ez a kamera (illetve az egész sugárkövetés) perspektívát is ábrázol, amely abban nyilvánul meg első látásra, hogy a távolabbi objektumok kisebbek, mint a közelebbi objektumok. Ezt átkapcsolhatjuk ortografikusra, és már teljesen más jellegű képet kapunk, amelyet inkább mérnöki ábrázoláshoz tudunk használni.

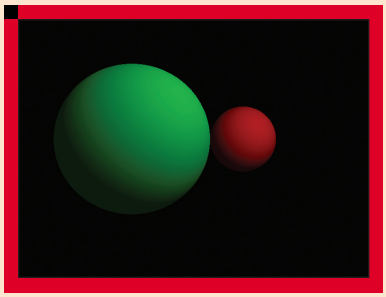
```
camera{
// orthographic
 location <0,0,-4>
 look_at <0,0,0>}
```

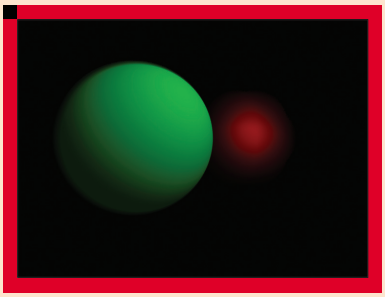
A *pov03.pov* által tartalmazott állományban az `orthographic` kulcsszó előtti komment jelet kitörölve-visszaírva a két vetítési mód között tudunk váltogatni. A vetítési módok között az a különbség, hogy a perspektíva esetén a végtelenben egy pontba futnak össze a nézetvonalak, az ortografikus esetén pedig a vetítősíkkal párhuzamos vetítővonalakkal számol a program. A két ábrán a vörös gömb mérete eltérőnek mutatkozik, holott a zöld és a vörös gömb mérete azonos, csak a vörös gömb hátrébb foglal helyet. A perspektívatorzítás miatt ezért kisebb mérete lesz (ezt szoktuk meg a valóságban is), mérnöki ábrázolásnál azonban nem szokás figyelembe venni a perspektívát, így a két gömb azonos méretűnek ***tűnik***. Mivel a *PoVRay* alapvető felhasználása a fotorealisztikus képek készítése, így a perspektíva az alapértelmezése, viszont vannak olyan mérnöki programok, amelyek az ortografikus ábrázolást igénylik.

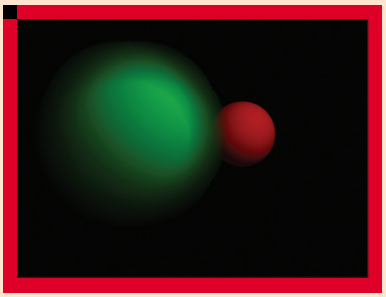
A perspektíva alapú kamera tovább állítható, például halszem optikát tudunk rátenni a kameránkra, amelynek az a lényege, hogy legalább 180° a látószöge. Eléggé furcsa élmény egy ilyen optika, viszont sok állat rendelkezik ilyen látással (a halakon kívül a madarak nagy része is), ugyanis ilyen szemmel mindent látnak, s ez a túlélésük tekintetében létfontosságú. *PoVRay* esetén a `fisheye` kulcsszó kell a kamera leírójába, s meg kell adni a szöget is, amely lehet akár 360° is (*pov04.pov*). További kameratípusok is találhatók a *PoVRay* eszköztárában, azonban ezek ritkán használatosak, némelyik pedig igen speciális célokat szolgál. Érdemes mindegyik vetítési módot kipróbálni, bár az eddig említett három mód tökéletesen elég az átlagos használathoz.

Ha megadunk egy kamerapozíciót és egy nézőpontot, akkor még alig határoztunk meg a kamerát. Kettő pont ugyanis egy vektort jelent csak, s meg kellene adnunk a kamera állását is, hiszen tarthatjuk akár fejjel lefelé is. Erre a célra szolgák a ***sky*** kulcsszó, amely a „felfelé” irányt adja meg. Bármilyen pontot meghatározhatunk a térber, a program ettől a ponttól egy merőlegest próbál húzni a már meghatározott szakaszra és ezzel a merőleges vektorral már meg tudja határozni a képsík pozícióját. Alapesetben az ***y*** koordináta lesz a képsík „felfelé” iránya.









A látószöget (***angle***) alapvetően meghatározza a kamera iránya (***direction***), s az a tény, hogy a képsík kiterjedése alapesetben egy egységnyi négyszög. Ha közvetlenül megadjuk a látószöget, akkor a képsík mérete nem fog megváltozni, az irányvektor hossza módosul. Ez gyakorlatilag azt jelenti, hogy az irányt még azelőtt kell meghatározni, mielőtt szöget adunk meg, mert látószöget módosítja, ha utána adjuk meg az irányvektort.
A képsík relatív méreteit az `up` és a `right` kulcsszó határozza meg, s alapesetben a képernyő felbontásához van „torzítva". A szélesség 4/3 szorzóval szélesebb, mint a magasság. Ha eltérő arányokat szeretnénk (például A4 méretű papírra vagy 10x15 fotópapírra dolgozunk), akkor például az

```
up y
right 1.5*x
```

sorokat kell a kamerában elhelyezni. Ez az arány minden esetben egyezzen meg azzal az aránnyal, amekkora aránnyal a képet elkészítjük, ellenkező esetben a képen torzulás fog jelentkezni. Ez főleg abban jelentkezik, hogy a gömbök vetülete nem kör lesz, hanem ellipszis.

Alapesetben kamera végtelen mélységélességgel rendelkezik, amely egy kissé ront a valóságérzeten, hiszen minden nem lehet egyszerre éles (hacsak nem lyukkamerát használunk :). Ezen a problémán tudunk segíteni a fókuszpont meghatározásával, amely általában a kamera nézőpontja is egyben. A fókuszpont környéke szokott éles lenni, s az élességtartomány méretét nevezzük mélységélességnek. Minél nagyobb ez a jelzőszám, annál nagyobb távolságot látunk élesnek. Ennek eléréséhez néhány egyéb beállítást is el kell végeznünk, amelyek közül az apertúra (***aperture***) jelenti a legfőbb paramétert. Ez tulajdonképpen a kamera optikáján található nyílás méretét jelenti, s a valósághoz hasonlóan a nagyobb átmérő kisebb mélységélességet jelent. Ha nem definiáljuk, akkor ennek az értéke nulla, amely végtelen mélységélességet jelent: minden pengeéles lesz, legyen az pár egységnyire vagy a távoli végtelenben. A megjelenítés minőségét a felhasznált fénysugarak számával (***blur_samples***) tudjuk befolyásolni: a 10 még elnagyoltnak és gyengének tűnik (de a kép készítése közben megfelelő kompromisszum lehet), 500 fölötti szám már feleslegesen szép eredményt ad. Természetesen minél nagyobb számot adunk meg, annál több időbe telik a kép elkészítése.

```
camera{
 location <0,0,-4>
 aperture 2.0
 blur_samples 500
 focal_point <0,0,0>
 look_at <0,0,0>}
```

A ***pov05.pov*** egyszerű másolata a harmadik példának, viszont a kamera esetén megadtuk a szükséges adatokat, amelyekből a program már képes mélységélességet számolni. A bal oldali kép esetén a zöld, s a jobb oldali esetén a vörös gömb került a fókuszba. Számítsunk arra, hogy a fotorealisztikus mélységélesség esetén – a példában is látható – két gömböt tartalmazó világ renderelése akár 1-2 percet is igénybe vehet egy korszerű (2-3 GHz-es processzorral szerelt) gép esetén!
A következő részben a világítás és a különböző fényforrások részletes beállításaival fogunk foglalkozni, illetve a program fontosabb parancssori kapcsolóival.

Auth Gábor
(auth.gabor@enaplo.hu)
Egy pécsi középiskolában informatikát és programozást oktat. Tíz éve botlott először a UNIX rendszerekbe, 7 év Linux használat után kapta el a FreeBSD lázat, amiből máig nem tudott kigyógyulni.

KAPCSOLÓDÓ CÍMEK

A PovRay projekt honlapja
➲ http://www.povray.org

Az SVG világa (1. rész)
Alapvetés, történet

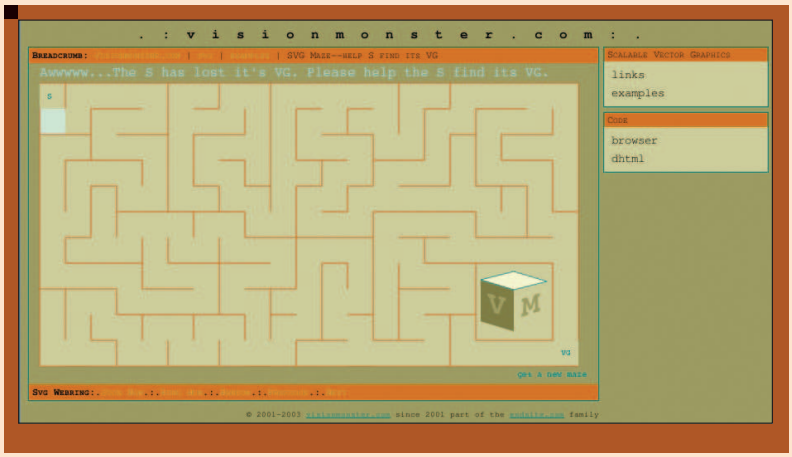
Most induló sorozatunkban az SVG-t fogjuk körüljárni. Tartson velünk, aki érdeklődik a webfejlesztés és a grafikus programok iránt, vagy ha egy kicsit szeretne a jövőbe tekinteni... Ízelítőként a tartalomból: W3C, MMS, XML, Sodipodi, Inkscape, Mozilla, Thunderbird, Konqueror, KDE, GNOME, PHP és így tovább. Ennek a hosszú felsorolásnak a tagjaiban az a közös, hogy mindegyik valamilyen kapcsolatban áll az SVG-vel. A lista az alkalmazások nagyon széles spektrumát jelenti a mobiltelefonoktól egészen a programozási nyelvekig. A sorozat első részében a W3C, az MMS és az XML titkairól lebben fel a fátyol.

Pixel- és vektorgrafika

Ahhoz, hogy az *SVG*-vel bármilyen szinten tudjunk foglalkozni, sőt, még mielőtt megtudnánk, mi ez egyáltalán, ismernünk kell a számítógépes grafikák két nagy csoportját. A raszteres képeket mindannyian jól ismerjük, ilyen képeket készít például a digitális fényképezőgép, a lapolvasó és a pixelgrafikus programok, például a *Gimp*. A sokféle részterület közös jellemzője, hogy az elkészült kép tovább nem nagyítható minőségromlás nélkül, valamint a kép méretével és színmélységével egyenes arányban nő a lemezen elfoglalt mérete, mivel elemi képpontokra felbontott képet tárol. Ezen tulajdonsága miatt gyakran a veszteségesen tömörített alfajaival találkozunk (*JPG* és *GIF*), ahol a mérete már kevésbé riasztó, de az állandó újratömörítések sokat rontanak a minőségen. A vektorgrafika ellenben alakzatokat és tulajdonságokat tárol, tehát csak a grafika összetettsége határozza meg a képméretet, a megadott információk alapján bármekkora kép kirajzolható, azaz tetszőlegesen nagyítható. Tipikus felhasználási területe a műszaki célú(például *CAD*) és a általános vektorgrafikus programok. Ha a webes felhasználást tekintjük, akkor a kettő előnyös kombinációja a legcélszerűbb. Jelen pillanatban ezt a legkiforrottabban a *Macromedia Flash*-sel tudjuk elérni. Ezen formátumok is néha tömörítettek, ellenben felépítésénél fogva veszteségmentesen tömörítik a vektorgrafikát. Az *SVG* esetében is tömöríthetjük a végeredményt, például a *gzip*-pel. Igaz, hogy ezt egy *BMP* (pixelgrafikus) képpel is megtehetjük, de az egyszerű szövegfájl, az *SVG*, jobban tömöríthető, mint a pixelgrafika. Ebben a környezetben tetszés szerint keverhetjük a raszteres és vektoros elemeket a leggyorsabb és leglátványosabb megoldások érdekében. De mi is a gond a *Flash*-sel? Előnye, hogy egész animációkat, valóságos kisfilmeket lehet vele gyártani, de nem nyílt szabvány, a web fejlesztőközpontjában, a *World Wide Web Consortiumnál* (a továbbiakban *W3C*) úgy vélték, szükség van egy teljesen platformfüggetlen, általános célú, szabványos vektorgrafikai formátumra, mely teljesen beleillik a többi *W3C* szabvány közé.

Az SVG története és fejlődése

Ez lett az *SVG*, a *Scalable Vector Graphics*, mely immáron hat éves múltra tekint vissza, hiszen 1999-ben jelent meg az első piszkozata a formátum leírásának. Nem kellett sok idő és az új specifikáció 2001-ben *W3C*



ajánlássá lépett elő ***SVG 1.0*** néven. Azóta frissebb változatok és új hajtások is kinőttek a ***W3C*** talajából, a tervek egészen szerteágazóak: a nyomtatásra szánt dokumentumoktól (***SVG Print***) a mobilalkalmazáson át (***SVG Mobile***) egészen a jobb szkriptelhetőségig (a ***Document Object Modellel*** való jobb együttműködés) mindenre van már javaslat. Ezeket a ***http://www.w3.org/Graphics/SVG/*** weboldalon magunk is szemügyre vehetjük.

A webes tartalom-előállítás (***Authoring Tool Guidelines***) és a fogyatékos látogatók számára készülő (***Accessibility Techniques***) elképzelések már igazán a jövőbe mutatnak, olyannyira, hogy még piszkozat sem jelent meg ezekről. Ami viszont a közeli jövőt illeti, az egyik friss újdonság az ***SVG***-s ***MMS***. A betűszó feloldása: ***Multimedia Message Service***. Eddig pixelgrafikus képeket, videókat küldözgethettünk, most viszont akár SVG fájlokat is.

Mi ebben a nagyszerű? Az ***MMS***-ért adatforgalom alapján kell általában fizetni, ezért nem mindegy, hogy mekkora az elküldött üzenet. A ***Nokia*** 2005 májusában adta ki az útmutatót (***http://svg.org/story/2005/7/5/181035/3748***). Mi sem természetesebb, hogy egyre-másra kerülnek az üzletek polcaira az ***SVG***-t ismerő mobiltelefonok, mint a ***Nokia 7710*** vagy a ***Sony Ericsson s600i***. Bár ezeknél a készülékeknél jogos a kérdés, hogy minek nevezzelek, mert gyakorlatilag mobiltelefonnal beoltott kéziszámítógépek.

A szabvány

Jelen pillanatban a legtöbb alkalmazás az 1.0-ás ***SVG***-t sem tudta teljes mértékben megvalósítani, az alkalmazásszállítók a szabványkiagyalók után kullognak ebben a tekintetben. A felhasználó számára olyan sok különbség nincs is az 1.0-ás és az 1.2-es formátum között, ellentétben például a szoftverek számozásával, ahol a végfelhasználó által látványos fejlesztések történnek. A fájlformátumok kidolgozása nem ilyen, az apró módosítások rejtve maradnak, minden az adott alkalmazáson múlik.

Térjünk rá most már magára a szabványra. Sok más formátumhoz hasonlóan az ***SVG*** is az ***XML 1.0***-ra (***http://www.w3.org/TR/2004/REC-xml-20040204/***) épül, amely mutat ugyan csekély hasonlóságot a webet jelen pillanatban is uraló ***HTML*** formátummal, de sokkal alkalmasabb programok általi kezelésre, mivel szigorú struktúrát követel meg. A ***HTML*** leírónyelvet ismerőknek például komoly változás, hogy minden címkénél(tag) kötelező a záróelem. Bővebben a (***http://www.w3c.hu/forditasok/XML_10_pontban.html***) ***W3C*** magyar oldalain olvashatunk arról, hogy az ***XML*** miben tér el a ***HTML***-től. Az ***SVG***-t a ***Flash***-hez hasonlóan kezdettől animációkra és interaktív működésre is felkészítették és éppúgy lehet benne raszteres kép is. Egy ***SVG*** dokumentum ehhez hasonlóan épül fel:

```
<?xml version="1.0"
standalone="no"?>
<!DOCTYPE svg PUBLIC "-
//W3C//DTD SVG 1.1//EN"

"http://www.w3.org/Graphics/SVG
/1.1/DTD/svg11.dtd">
<svg width="2cm" height="2cm"
version="1.1"
xmlns="http://www.w3.org/2000/
➥svg">
  <desc>Egy negyzet
  </desc>
    <rect x="0.5cm" y="0.5cm"
width="1cm" height="1cm"/>
</svg>
```

Amint a <desc> tagek között megadtuk, ez nem tesz mást, mint megjelenít egy négyzetet. Aki az ***XHTML*** szabványt ismeri, annak ismerősek a szerkezeti elemek, hiszen a technikailag fejlett weboldalak ma már a ***HTML 4*** helyett az ***XHTML***-t részesítik előnyben. Habár a példát bármilyen szövegszerkesztővel elő lehet állítani, az ***SVG*** összetett nyelv, csak ***SVG***-szerkesztővel érdemes nekivágni a komolyabb munkának. Aki mégis szeretné ***XML*** szinten beleásni magát az ***SVG***-be, annak az egyszerű szövegszerkesztő helyett a ***KXML Editor*** (***http://kxmleditor.sourceforge.net/***) nyújt segítséget. Ezzel a szoftverrel bármilyen ***XML*** formátumú fájlt megjeleníthetünk többféle struktúrában és szerkeszthetjük is.

Ebben a példában csupán egyetlen grafikát leíró taget próbáltunk ki, a rect-et, így most tekintsük át, mi mindent lehet ***SVG***-ül mondani. A legfontosabb, hogy a ***CSS 2***-ből megismert formázások többnyire alkalmazhatók az ***SVG*** elemekre. De melyek is ezek az elemek? Lehet négyszöget, kört, ellipszist, sokszöget és törtvonalat rajzolni. Alapvetően ezekből a geometriai elemekből és a tetszőleges kombinációjukból állnak a bonyolultabb ***SVG*** rajzok is. Emellett használhatunk szöveget, különféle átmeneteket, szűrőket és átalakításokat végezhetünk, forgathatjuk a koordinátarendszert például.

A nyelv nagyon sokrétű, így ahhoz, hogy megállapítsuk egy fájlról, hogy van-e benne hiba, a ***W3C*** kiadta az ***SVG Validatort***, mellyel egyetlen paranccsal kiderül, követi-e a ***W3C*** specifikációját az adott fájl. Ezt a hasznos ***Javaban*** írt(azaz szinte minden platformon működő) programot a ***http://jiggles.w3.org/svgvalidator/*** címről tudjuk letölteni illetve ugyanitt on-line is kipróbálhatjuk.

A gyakorlat

Most, hogy egészen a mélyére néztünk az ***SVG***-nek, váltsunk perspektívát és nézzük meg, hogy ez az elméletben jól

csengő formátum hol tart a gyakorlatban. Ma már szinte mindegyik webböngésző felkészíthető az *SVG* fájlokra, azonban a *Macromedia Flash*-sel ellentétben az elterjedtsége csekély, nagyon kevés böngészőben van beépített támogatás, az *SVG* bővítmények (plugin) pedig nem terjedtek el. Ráadásul a natív *SVG* támogatás a böngészőkben gyenge minőségű, *Firefoxszal* (1.0.4-es változat) a szkriptelhető *SVG* nem ment, a *Konqueror* (3.3.2) ugyan támogatta, de sok esetben rövid használat után összeomlott. Érdemes mégis foglalkozni az egésszel annak ellenére, hogy jelen pillanatban ilyen csekély az *SVG* elfogadottsága? Feltétlenül. Az *Adobe* több platformra is fejleszti az *SVG* pluginjét és *SVG*-nézegetőjét és minden magára valamit is adó webböngésző gőzerővel dolgozik az egyre tökéletesebb natív *SVG* támogatáson. Olyan komoly grafikus alkalmazások és cégek sorakoztak fel az *SVG* mögött, mint a *KDE*, a *Gimp*, a *Mozilla Foundation*, a *Gnome*, az *Adobe Illustratorja* és még sokan mások. Elsősorban a webes használat tör előre, azonban *Gnome* és *KDE* esetében már a rendszeren belüli használat is lehetséges.

Látványos SVG a weben

A *http://www.carto.net/papers/svg/samples/* címen számtalan látványos *SVG* képet, animációt és interaktív alkalmazást találunk, melyek nagyon jól szemléltetik, hogy mire is alkalmas már jelen pillanatban az *SVG*. A példák többségéhez nem elegendő a böngészők beépített *SVG* támogatása, érdemes beszerezni az *Adobe* által kibocsátott *SVG* nézegetőt (*http://www.adobe.com/svg/viewer/install/main.html*), mert így minden korlátozás nélkül úgy tudjuk megnézni az összetett *SVG* animációkat, ahogy azt a szerző kitalálta. Az *Adobe* csupán egy *Red Hat Linux* alatt használható bétát adott ki, de szerencsére általános célú telepítőprogramja van, így bárhova feltelepíthető. Elméletileg a letöltött *.tar.gz* fájlból kitömörítés után csupán az *install.sh*-t kell elindítani, de *Debian Sid* alatt nem működött, viszont kézzel végrehajtva a fájlban lévő utasításokat, kiválóan be lehetett üzemelni és életre keltek *Firefoxban* a halott *SVG*-animációk is. Azonban jó tudni, hogy az *Adobe SVG* megoldásai zárt forráskódúak.

SVG, WEB és HTML?

Ha a weboldalainkon *SVG*-t akarunk használni, akkor ugyan használhatjuk a *HTML 4*-es leírónyelvet is, azonban jó ha tudjuk, hogy ahogyan grafikában *XML* alapú formátumot használunk, ezt megtehetjük az egész weboldalon is. Az *SVG*-t támogató böngészők úgyis követik a friss szabványokat, így aztán nem veszítünk látogatót, ha magát az oldalt sem elavult módon építjük fel. Mivel ez a cikksorozat alapjában véve az *SVG*-ről szól, részletesen nem foglalkozunk az *XHTML*-el, de magyar nyelven (*http://htmlinfo.polyhistor.hu/xhtml1se/cover.html*) is utánajárhatunk, hogyan kell *XHTML* oldalakat írni. Kicsit ugyan mellékszálnak tűnhetnek a fentiek, de amennyiben az *SVG*-t a webfejlesztés szemüvegén keresztül nézzük, kikerülhetetlen az a kérdés, hogy magát az oldalt, ahova a grafika kerül, hogyan készítsük el.

Átjárás az alkalmazások között

Ha általánossá válik a vektorgrafikus programok körében az *SVG* támogatás, egy közös nevezővé válik ezen alkalmazások között, hiszen egy nyílt formátum, melyet bárki megvalósíthat szoftverében, nem gátolják jogi problémák. A másik eshetőség, hogy minden nagy szoftvergyártó csavar egy kicsit a formátumon és egymást csak hellyel-közzel értő részekre esik szét az *SVG*-t használók tábora. Jelen pillanatban a linuxos grafikai programokat kipróbálva szerencsére az első változat látszik megvalósulni, bonyolultabb grafikákat betöltve, mentve a sok különféle *SVG* alkalmazás között semmi probléma nem bukkant föl. Ráadásul már most is lehet például *AutoCadből SVG*-be menteni, ami tovább erősíti azt az elképzelést, hogy a csereformátum szerepét is betölti az új szabvány.
Idevágó költői kérdésünk: lehet ennél még kényelmesebb is? Azon túl, hogy a vektorgrafikus programok egymás közt remekül eltársalognak *SVG*-ül, a vektorgrafikát egy csapásra pixelgrafikussá alakíthatjuk, ha a helyzet úgy kívánja. A *GIMP* ugyan elsősorban raszteres képekkel dolgozik, de az *SVG*-t be tudja olvasni és onnantól kezdve a *GIMP* által támogatott összes formátumban elmenthetjük a képet.
Remélhetőleg a közeljövőben, amikor nem csak a szabad szoftver közössége fogja felismerni az *SVG* jelentőségét (azért már ma is vannak kivételek), akkor is megmarad ez az átjárhatóság. Mivel az *SVG*-hez számtalan egyéb webes formátum kapcsolódik (*XHTML*, *CSS 2*, stb), ha elkezdődik például az *Internet Explorerben* az *SVG* támogatása, az új webes technológiákkal is lépést fog tartani a most kissé elmaradt (nem felhasználószámban, hanem szabványkövetésben) böngésző.
A következő részben már *SVG*-t támogató alkalmazásokkal fogunk dolgozni! Akár *Linuxot*, akár valamelyik *BSD*-t vagy esetleg *Windowst* használjunk is, kényelmes *SVG*-szerkesztőkkel fogjuk gazdagítani webes fegyvertárunkat! Addig is figyeljük mobilszolgáltatónk kínálatát, mikor tűnik fel az első *SVG*-megfelelő mobil *Magyarországon*!

Novák Áron
(aaron@szentimre.hu)
BME-VIK-es gólya, működvelő rendszergazda. Jelenleg leginkább a NetBeans-szel és mindenféle hordozható eszközzel foglalkozik, legalábbis mindazokkal amelyeket meg lehet szólaltatni Linux alatt.

KAPCSOLÓDÓ CÍMEK

SVG példaoldalak:

- ➲ www.carto.net/papers/svg/samples/
- ➲ www.w3schools.com/svg/svg_examples.asp
- ➲ www.croczilla.com/svg/samples/
- ➲ www.w3.org/Graphics/SVG/Test/ – a hivatalos tesztoldal
- ➲ www.taedium.com/svg/
- ➲ www.visionmonster.com/svg/ – egy egész oldal SVG-ben

SVG leírások:

- ➲ www.w3.org/TR/SVG/ – a hivatalos leírás
- ➲ www.w3schools.com/svg/default.asp
- ➲ svg.lap.hu/

Kell egy Csapat! Munkacsoportok támogatása – nem csak Linux alatt

Csapatmunka. Vajon csak egy napjainkban divatossá vált kifejezés, vagy komoly háttér van mögötte? Én úgy vélem mindenképpen az utóbbi állítás az igaz.

Aki valaha is próbált már két embernél többel egyszerre együtt dolgozni, az tisztában azzal, hogy mi jelent a jó szervezés, az információk hatékony megosztása. Évekkel ezelőtt ilyen célra hirdetőtáblákat használtak, körtelefonok mentek egész nap a csapat tagjai között. Aztán megjelent az internet, az e-mail, a ***World Wide Web***. Végre lehetett egyszerre hatékonyan sok emberrel információt megosztani, megjelent egy olyan környezet, amelyet hatékonyan lehetett használni arra, hogy megszervezzük a munkát, megszervezzük az életünket. Több cég felismerte az igényt és a 90-es évek elejétől sorra jelentek meg a csoportmunkát támogató megoldások. ***Lotus***, ***Exchange***, hogy csak a két legnagyobbat említsem.
A csoportmunkát támogató rendszerek mindegyiket törekedett arra, hogy amellett, hogy az egyén megszervezheti a saját munkáját, mindezt oly módon tegye, hogy azzal közvetve a többiek munkaszervezését is befolyásolja, segíteni tudja. Legelső elterjedt megoldások a közösen használható naptárak, névjegyalbumok voltak – természetesen az integrált levelező rendszer mellett.
Az első csoportmunka rendszerek megjelenése óta eltelt időben az informatika és a távközlés olyan fejlődésen ment keresztül, amely alapvetően átalakította mind a lehetőségeket, mind pedig a felhasználók igényeit. Technológiai oldalról elképesztő újítások jelentek meg. A tenyér méretű számítógép, a mobiltelefon, amely rendelkezik olyan hatékony kommunikációs hálózattal, amellyel olcsón és gyorsan lehet adatot átvinni, megjelentek olyan hálózati protokollok, amelyeken keresztül több gépet tudunk kezelni, állandóan szinkronban tudjuk tartani az adatainkat. Nézzük hát, hogy mit is várhatunk és mit kapunk egy ilyen rendszertől.

Naptár

A két legnépszerűbb és leggyakrabban használt szolgáltatás egyike a közösen is használható határidő naplók elektronikus reinkarnációja. Szinte minden csoportmunka alkalmazás tartalmaz ilyen szolgáltatást – legalábbis azok, amelyek komolyan gondolják, hogy felhasználókat fognak kiszolgálni.
Az ilyen ***naptárakat*** használhatjuk arra, hogy saját személyes bejegyzéseket helyezzünk el, oly módon, hogy ezt más felhasználók ne lássák, ugyanakkor készíthetünk olyan bejegyzéseket is, amelyekhez más felhasználókat is hozzárendelünk. Ez egy nagyon hasznos funkció, ha szervezünk egy olyan megbeszélést, ahol egy csapatból több ember is jelen lesz, ilyenkor az időpont és helyszín egyeztetése után a szervező a résztvevők naptárában is elkészítjük a bejegyzést, sőt az okosabb rendszerek e-mailben, SMS-ben értesítik a résztvevőket, hogy új meghívót kaptak egy eseményre. A meghívókat el lehet fogadni, ezzel visszaigazolhatjuk a részvételt, el tudjuk utasítani, ha valamilyen ütközés volna egyéb bejegyzéssel. Ma már vannak olyan naptárkezelő rendszerek, amelyek nem csak emberi erőforrásokat képesek kezelni, így nem csak a résztvevőket vehetjük fel egy adott időponthoz, hanem foglalhatunk vele tárgyalót, projektort, vagy egyéb más erőforrást. Ez a megoldás sok környezetben megkönnyíti a mindennapi munkaszervezést, könnyen elkerülhetjük azokat az eseteket, amikor többen is egy időpontban mennének be egy tárgyalóba, vagy éppen a parkolóban keresi az ember a céges autót és azon bosszankodik, hogy ki vitte el már megint.

Névjegyek

A másik nagyon népszerű szolgáltatása a csoportmunka rendszereknek a ***névjegyalbum*** szolgáltatás. Ezek az alkalmazások teljes mértékben helyettesítik, sőt kiterjesztik a korábbi névjegykártya gyűjtők szerepét. Az egyes bejegyzéseket a felhasználók egyedül, vagy közösen kezelhetik, követhetik a változásokat és mindig rendelkezésére áll azoknak az embereknek, akik a rendszert használják. Napjainkban azt hiszem elmondható, hogy a mobil-

telefonok terjedésének egyik következménye, hogy az emberek egyre kevesebb telefonszámot jegyeznek meg, hiszen mindenki aki fontos, név alapján benne van a telefonunkban. Innen már csak egy lépés annak megoldása, hogy az irodai, vagy az otthoni számítógépünk a fontos névjegyeket a telefonunkra küldje, így egy-egy névjegybejegyzés módosítása után biztosak lehetünk abban, hogy a névjegyet használók mindig a legfrissebb adatbázissal dolgoznak. Ennek megoldására mostanság már egyre több rendszer és mobil eszköz támogatja a ***SyncML*** protokollt, amely pontosan ezt a feladatot hivatott elvégezni az Interneten keresztül.

A naptár modulhoz némileg kötődő funkció a ***teendők*** feljegyzésére szolgáló modul, amely szintén mindegyik rendszerben megtalálható. Ez némely programoknál nem jelent többet, mint olyan bejegyzések, amelyek emlékeztetőül szolgálnak bizonyos feladatok elvégzéséhez, azonban az okosabb szoftverek ezt a funkciót is jelentősen továbbfejlesztették. Vannak olyan rendszerek, ahol egy-egy bejegyzéshez erőforrásokat rendelhetünk, a feladat elvégzésére időintervallumokat adhatunk meg, prioritásokat rendelhetünk az egyes funkciókhoz, sőt akár követhetjük, hogy a feladatot az adott napig milyen részben készítettük el. Ezekkel a funkciókkal és némi szervezéssel akár egyszerű projektirányító rendszerré is válhat a teendők modul.

Levelezés

Talán minden rendszer alapja, a legfontosabb funkció, amely nélkül az egész rendszer nem sokat ér. Szállítótól függően más és más megoldással találkozhatunk ezen a téren is. Vannak szállítók, akik saját, beépített levelező megoldással szállítják a rendszerüket, ilyenre a legegyszerűbb példa a Microsoft Exchange. Megint más szállítók azt a taktikát követik, hogy az egyszer már elkészült és rendelkezésre álló modulokat nem valósítják meg újra, hanem használják, főleg ha az ingyenesen, szabadon rendelkezésre áll. Ilyen meggondolás alapján építik a Linuxos munkacsoport kezelő szoftverek jelentős részét, így ki lehet aknázni a Linuxban meglévő meglehetősen erős és jól használható levelező szolgáltatásokat.

Wiki

Ejtsünk néhány szót az internetes tudás egyik legújabb gyűjtő, tároló és megjelenítő formájáról, a ***Wiki***-ről. A ***Wiki*** szó eredete a Hawaii ***wiki wiki*** kifejezésből ered, ami körülbelül annyit tesz, hogy ***gyors***, ***közvetlen*** (forrás: ***wikipedia.org***). Az internetes jelentése a szónak egy olyan interaktív tudástárat fed, ahol a felhasználók közösen készítik el az adott oldalon megjelenő információt, mégpedig olyan módon, hogy az adott oldalakat minden felhasználó saját szája íze és legjobb tudása alapján tudja szerkeszteni. Mára az egyik legnagyobb adatbázissal rendelkező internetes enciklopédia is ezen a merőben újszerű alapon kezdte meg működését. Aki nem ismerné, annak ajánlom figyelmébe a ***wikipedia.org*** című oldalt.

A ***wiki*** tehát gyorsan közkedvelt eszközzé vált a tudás orientált környezetekben, így nem csoda, hogy a munkacsoportos rendszerek készítői is felfigyeltek rá és mára több rendszer elhagyhatatlan moduljává vált. Egy-egy projekt kapcsán nagyon jól használható, amikor több ember közösen gondolkodik és dolgozik egy adott témán- Látványosan és jól használhatóan tudják egymás tudását közös tudássá tenni, így nagyon hatékony eszköz a használható tudás létrehozása terén akár vállalati-, akár oktatási-, vagy éppen magáncélokra.

Visszakeresés

Természetesen a csoportos munka semmit nem érne, ha a munka során elkészített dokumentumainkat nem tudnánk rendszerezni, elhelyezni egy megfelelő, később visszakereshető helyen. A tárgyalt rendszerek többsége erre is kínál valamiféle megoldást. Vannak rendszerek, amelyek ilyen célra külön, csak a rendszerben elérhető mappastruktúrákat kínálnak. Ebben az esetben a rendszerben a létrehozott mappákra megadható hozzáférési jogosultság, amely a felhasználók, vagy felhasználói csoportok számára deklarálja az adott anyaghoz való hozzáférés szabályait. Ez egy nagyon hasznos dolog, hiszen egy nagyobb projekt kapcsán meglehetősen sok embernek kell együtt dolgoznia, ugyanakkor a felhasználók általában meglehetősen heterogén hozzáférési jogokkal rendelkeznek. Így egy a projekt felügyeletét végrehajtó felhasználó személy például olvasási jogokkal hozzáférhet a teljes rendszerhez, míg egy adott terület felelőse csak az adott terület dokumentumait kezelheti, azokat viszont teljes jogosultsággal.

Egy másik szemlélet szerint a rendszer nem kínál külön mapparendszert a dokumentumok tárolására, ellenben hozzáférést engedélyez egy meglévő fizikai állományrendszerhez. Ennek a megoldásnak előnye, hogy a dokumentumok nem csak az adott rendszeren keresztül, hanem attól függetlenül is elérhetőek, ám ez a funkcionalitás van amikor éppen hátrány, nem pedig előny.

Megint más rendszerek egészen odáig mennek, hogy nem csak egy mappastruktúrában tárolják az adatokat, hanem gyakorlatilag egy adatbázisban, méghozzá megjegyzésekkel ellátva a dokumentum teljes, vagy kivonatos tartalmával. Ezen metaadatok segítségével a későbbiekben a dokumentumok az előző megoldásokhoz képes sokkal hatékonyabb módon lesznek visszakereshetőek. Az ilyen rendszerek hátránya jelenleg talán a kicsit körülményes feltöltés, hiszen nem elég magát a dokumentumot a helyére tenni, azt megfelelő metaadatokkal is el kell látni. Ugyanakkor a jövő mindenképpen ebbe az irányba mutat, hiszen ezzel a módszerrel olyan információ állítható elő, amelyet aztán a későbbiekben ember és gép számára is tudássá lehet alakítani.

Fórum

Ha már a ***wiki*** rendszerektől ejtettem szót, akkor említsük meg a ***wiki*** közeli rokonát, a ***fórum*** szolgáltatást is. A korszerűbb munkacsoportos rendszereknek természetesen ez a modul is jóformán elválaszthatatlan részét képezi. A ***wiki***-vel ellentétben itt nem egy dokumentum formálása a cél, hanem sokkal inkább a párbeszéd dokumentálása, amelynek kapcsán a végén egy döntés, vagy dokumentum megszületik. A ***wiki*** és ***fórum*** nagyon jól kiegészíti egymást az interaktív tudás létrehozása terén.

Kapcsolódás a rendszerhez

Most, hogy végére értünk a fontosabb funkciók áttekintésének, nézzük meg, hogy milyen módon tudunk egy ilyen munkacsoportos rendszerhez kapcsolódni.

Azoknál a rendszereknél, amelyek története a 90-es évek elejére nyúlik vissza – mint az ***Exchange***, vagy a ***Lotus*** – szinte mindenütt találunk a rendszerhez kapcsolódó natív kliensprogramot. Nem meglepő a dolog, hiszen a 90-es évek elején ezek a rendszerek a vállalati hálózatokon üzemeltek és a meglévő hálózati és rendszer infrastruktúrán kellett a működést megvalósítani. Ennek legegyszerűbb módja az akkori felfogás szerint egy kliens-szerver alapú program volt. Így jelent meg a Microsoft kínálatában kiszolgáló oldalon az ***Exchange Server***, míg kliens oldalon az Office csomag és benne az ***Outlook*** nevű kliens program.

Ezután a 90-es évek második felében az Internet óriási fejlődésen ment keresztül, megjelentek a szerver oldali programozási nyelvek és rohamosan terjedt a dinamikus tartalmat megjelenítő web. Nem is csoda, hogy nem kellett sokat várni és megjelentek az új irányzatot képviselő, immár internetes technológiákon alapuló rendszerek, valamint a korábbi rendszerek készítői is gőzerővel dolgoztak azon, hogy rendszereik az internet felé is képesek legyenek adatot megjeleníteni.

Az elmúlt években megjelenő rendszerek szinte kivétel nélkül azt a stratégiát képviselték, hogy egy nagyon fejlett webes alapú megjelenítést tartalmaznak és ehhez a rendszerhez lehet megfelelő interfészek felhasználásával natív klienseket csatolni. Így történt az, hogy a legújabb rendszerekhez már akár a méltán népszerű ***Evolution***, vagy ***Mozilla Thunderbird*** kliensekkel is tudunk csatlakozni. Megfigyelhető ugyanakkor, hogy a munkacsoportos rendszerek használóinak egy jelentős része a natív kliens használata helyett kezd inkább átállni a webes kliensek használatára. Nem véletlen, hogy a különböző webmail rendszerek óriási népszerűségnek örvendenek, hiszen ezek olyan felületen nyújtanak hozzáférést a postaládánkhoz, amely a világon bárhonnan, beállítások elvégzése nélkül is teljesen biztonságos módon működik.

A webes technológia alkalmazásának másik nagy előnye, hogy egyre több rendszer kínál napjainkban a felhasználóknak olyan szolgáltatást, amellyel könnyen és gyorsan ki tudják alakítani internetes otthonukat, így egy vállalati rendszeren belül a megfelelő alkotóelemek felhasználásával összetehetnek egy olyan kis privát oldalt, amelyen az általuk legfontosabbnak ítélt információk érhetőek el, a számukra legfontosabb dolgok kerülnek megjelenítésre. Amennyiben ezt a megoldást egy úgynevezett ***WYSIWYG (What You See Is What You Get – amit látsz, azt kapod)*** megoldással tudják kombinálni, akkor ténylegesen egy olyan hatékony eszköz áll a rendelkezésünkre, amellyel csak össze kell dobálni a kívánt oldalt.

Most hogy végére jutottunk a funkcionális bemutatásnak és megemlítettünk egy-két nagy nevet a piacról nézzük meg, hogy ***Linux*** alatt, lehetőleg nyílt forráskódú alapokon milyen rendszerek állnak rendelkezésünkre. Mert abban ugye senki nem kételkedett, hogy vannak ilyenek...

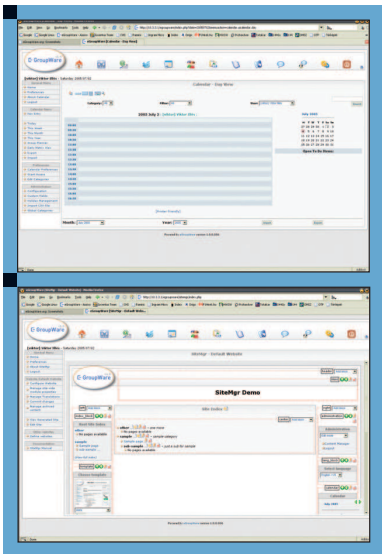
Én három rendszert szeretnék röviden bemutatni, az ***eGroupware***-t, az ***OpenGroupware***-t, valamint az ***OpenXchange***-et. Az, hogy én ezt a hármat választottam nem azt jeleni, hogy más nincs, mindössze én ezzel a három rendszerrel kerültem közelebbi ismeretségbe, ám aki komolyan gondolkozik egy hasonló rendszer bevezetésén annak ajánlom, hogy térképezze fel rendesen a kínálatot és válassza ki azt a rendszert, amelyik a saját igényeket a legteljesebben kielégíti.

Az eGroupware

Vizsgálatunk első alanya egy ***PHP*** nyelven írt rendszer. Akinek esetleg a rendszer nagy hasonlóságot mutatna a ***PHPGroupware*** nevű rendszerhez, az ne csodálkozzon, a projekt gyökerei azonosak, a két rendszer funkcióiban szinte teljesen megegyezik.

A rendszer telepítése nagyon egyszerű, szinte mindegyik Linux disztribúcióhoz megtalálható bináris telepítő készlet, így sokat nem kell vele bajlódnunk. Én ***Debian Sarge*** alá telepítettem, gyakorlatilag az ***apt-get install*** mechanizmust használtam csak. A rendszernek szüksége van egy adatbázis kezelő rendszerre a háttérben, egy webszerverre és természetesen a hozzá kapcsolódó ***PHP*** modulra. Az adatbázis kezelő terén a rendszer támogatja mind a ***MySQL***, mind pedig a ***PostgreSQL*** adatbázis kezelőket, így semmiképpen nem fog gondot jelenteni a rendszer telepítése.

Nagyon jól használható megoldás az ***eGroupware***-be beépített ***LDAP*** alapú



azonosítás, így aki rendelkezik ***LDAP*** kiszolgálóval, az nagyon egyszerűen ki tud alakítani olyan rendszert, amely megfelel az úgynevezett ***Single Sign On*** filozófiának.

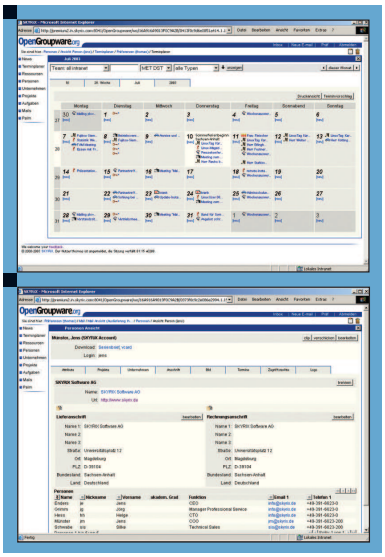
A rendszer gyakorlatilag a fent felsorolt összes modult tartalmazza, sőt annál még többet is. Nagyon jól konfigurálható, használata rendkívül egyszerű, bár a megjelenítése talán egy kicsit lassú. Elvileg támogatja a naptár megosztását a ***WebDAV*** protokollon keresztül, bár bevallom én ezt a szolgáltatását a rendszernek még nem használtam, így annak pontos működéséről és használhatóságáról nem tudok nyilatkozni.

Jelenleg a rendszer 1.0.007-es verziónál tart és ***stable*** állapotban van, így üzleti, vagy egyéb kritikus környezetben is biztonsággal használható. Erőforrásigénye 40-50 felhasználóig nem haladja meg egy PC alapú szerver korlátait, de ha úgy éreznénk, hogy a felhasználók hajlamosak túlterhelni a rendszert, akkor a webkiszolgáló megfelelő beállításával megvédhetjük a rendszert az összeomlástól.

A projekt megtalálható a ***www.egroupware.org*** weblapon és itt egy demo oldal is található annak, aki kipróbálná a rendszert működés közben is.

Az OpenGroupware.org projekt

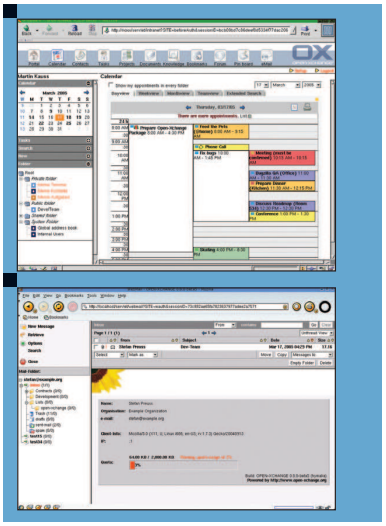
A mustra második alanya az ***OpenGroupware***. Az ***OpenGroupware*** egy német termék, a ***Skyrix*** nevű munkacsoportos alkalmazáscsomag nyílt forráskódú változata. A rendszernek tehát van egy teljes értékű, de fizetős



változata, így jogosan merül fel a kérdés, vajon mennyit tud és mennyire megbízható a nyílt forráskódú változat. Ha vetünk egy pillantást a termék jelenlegi verziójára (1.0a), akkor rögtön kiderül, hogy ez még csak egy alfa verzió, ám a fejlesztők állítása szerint már egy működő rendszerről van szó. Hogy ezt bizonyítsák, az oldalról letölthető egy úgynevezett *LiveCD*, amit kiírva egy üres korongra és ezzel indítva a gépet egy olyan rendszer áll fel, amelyen egy előre beállított *OpenGroupware* fut. Ez a megoldás számomra nagyon szimpatikus volt, úgy próbálhattam a ki a rendszert, hogy nem kellett egyik gépemet sem „széthegeszteni" és utána napokig azon fáradozni, hogy visszaállítsam az eredeti állapotokat.
A rendszer, mint mindegyik tárgyalt program, természetesen rendelkezik egy webes felhasználói felülettel, amely funkcióiban teljesen hasonló az *eGroupware* rendszerhez. Amiben az *OpenGroupware* nagyot lépett előre az az, hogy a rendszer rendelkezik egy *Outlook*/*Evolution* csatolóval is, így natív klienssel is teljes értékűen használható rendszer készül a fejlesztők keze alatt. Végignézve a projekttel kapcsolatos fejlesztéseket, nagy örömmel láttam, hogy *SyncML* interfész is készül az *OpenGroupware*-hez, így ha kiadásra kerül a stabil verzió, akkor azt hiszem, hogy a Microsoft felkötheti a nadrágot, mert egy olyan konkurens fog a piacra lépni, amely szinte azonos, sőt sok esetben még jobb szolgáltatásokat is fog nyújtani, mindezt nyílt forráskódú alapokon.
Már őszintén várom, hogy kipróbálhassam az első stabil verziót...
A projekt megtalálható a *www.opengroupware.org* oldalon.

Az Open-Xchange projekt

A harmadik alany egy szintén nagyon biztató projekt terméke, annak ellenére, hogy a három vizsgált rendszer közül ez van a legkorábbi fejlesztési fázisban. Az *Open-Xhange* egy *Java* alapú rendszer, jelenleg 0.8-as verziónál tart. Annak ellenére, hogy a telepítéséhez *Debian* alá nagyon korrekt, lépésről lépésre magyarázó útmutató van, nem volt egy könnyű dolgom vele. Sajnos a rendszer építőelemei nem érhetőek el *Debian* csomagok formájában, így azokat egyesével kellett leszedni a hálózatról, egyesével kellett őket telepíteni, amitől az *aptitude* világában őszintén szólva kicsit már elszoktam. De a jó dolgokért meg kell szenvedni. És az *Open-Xchange* egy nagyon jó program és jó belegondolni, hogy milyen jó lesz, ha kijön az első stabil verzió.
Annak ellenére, hogy *Java* alapú és a háttérben egy *Tomcat* futott, meglepően fürge volt a rendszer, őszintén szólva rosszabbra számítottam.
Adatbázis kezelőnek a telepítési útmutatónak megfelelően *PostgreSQL*-t használtam, amit amúgy is jó választásnak tartok, így itt sem csalódtam benne. A rendszerhez telepíteni kellett még egy Java *SDK*-t, Apache *Tomcat*-et és néhány kapcsolódó csomagot.
Nagyon pozitív, hogy a leírás annyira részletes, hogy tényleg csak azzal kell foglalkozni, hogy mindent a leírás alapján végezzünk el, akkor nem kerülhetünk bajba.
A három vizsgált rendszer közül őszintén szólva nekem ez tetszett a legjobban. A felülete nagyon letisztult, egyszerű, könnyen átlátható, remélem ezen nem változtatnak.
A rendszer itt is tartalmazza a fent leírt modulokat, ebben sem kell csalódnunk. Tartalmaz a szoftver továbbá egy webmail klienst is, ami nekem kifejezetten tetszett. A használata pofon egyszerű, az oldal elrendezése nagyon jól átláthatóvá teszi a programot, úgyhogy tényleg csak a legjobbakat tudom róla elmondani.
Ami a rendszer elérését illeti, örömömre szolgált, hogy itt is tervbe van véve a natív kliensek támogatása, így a jövőben várhatóan megjelenik egy *Outlook*/*Evolution* interfész, amely biztosan nagyon felkapottá teszi majd a rendszert. Aki kíváncsi a rendszer működésére és nem szeretne a telepítéssel vesződni, annak ajánlom figyelmébe, hogy a projekt weblapján itt is található egy online demó. A teljes projekt elérhető a *www.open-xchange.org* címen.



Összefoglalás

Nagy örömömre szolgált, hogy a nyílt forráskódú rendszerek a szoftverpiac egy újabb szegmensében próbálják megvetni a lábukat – minden bizonnyal sikerrel. A látottak bizakodással töltenek el, hogy pár hónapon belül egészen biztosan több igazán jól működő rendszer körül válogathatunk ha ilyen megoldásra lenne szükségünk. Sajnos azonban el kell mondani, hogy jelenleg az egyetlen igazán használható megoldás a három rendszer közül az *eGroupware* rendszer, a többit még a fejlesztők sem ajánlják mindennapi használatra. Ám a közeljövőben a helyzet egészen biztosan megváltozik, így mindenkit, akit érdekel a téma, csak arra tudok buzdítani, hogy kísérje figyelemmel ezeknek a rendszereknek a fejlődését, mert a jövőben alternatívát fognak jelenteni a jelenleg használt rendszerekkel szemben.

Illés Viktor (viktor@ei.hu)
A BME műszaki informatikus szakának hallgatója, mellette weblapokkal, linuxos és windowsos rendszerekkel foglalkozik.