Beköszöntő

Linuxvilág

A szerkesztői tevékenységnek van néhány érdekes „mellékhatása”. Az egyik ilyen a gyakori időutazás. Az hagyján, hogy az ember jó egy hónappal korábban olvashatja a ***Linux Journal*** anyagát, mint a „többiek”. Az már sokkal érdekesebb, hogy ma reggel, mikor állandó szerzőtársammal, ***Kuksi professzorral*** sétáltunk, még esett a hó, de én már körülbelül tudtam, milyen cikkek jelennek majd meg a ***Linuxvilág*** júniusi számában. Szintén érdekes érzés a „közszereplővé” válás, ami a lapkiadásnak ugyan természetes következménye, mégis meglepetéseket okozhat a gyanútlan szerkesztőnek helyenként és időnként. Aki ugyanis középen áll, az – ha akarja, ha nem, ha számított rá, ha nem – egyszerre kap csendesen elmondott jó tanácsot és korlátolt bekiabálást, egyszerre láthat szakmai tudással párosult szerénységet és gondolatilag zavaros ám jól fejlett kispolgári öntudatot. Ezt a zűrzavart hívják tömören demokráciának, amelynek a végkifejlete alapvetően attól függ, mennyire képes megőrizni az egyensúlyát.

No, de vissza a jelenbe, a márciusi számhoz.

A ***Magazin*** rovatban ***Novák Áron*** a Wiki jelenségről ír, míg ***Rácz Zoltán*** a „kernelmagyarítási projekt” történetéről és előrehaladásáról számol be. A ***Fejlesztői sarok*** ebben a számban ***Tóth Péternek*** és ***Radics Péternek*** jutott. Előbbi szerzőnk az ***OpenGL*** programozással foglalkozó sorozatot folytatja, míg utóbbi az ***XML*** rejtelmeibe kezdi bevezetni az olvasót. Napjaink egyik slágertémája a virtualizáció, ami kicsit leegyszerűsítve olyan szoftveres megoldásokat takar, melyek segítségével az egyik operációs rendszer a másikon futtatható. E téren az egyik első, és ma is igen népszerű megoldás a ***VMware*** céghez fűződik, amely nemrég ***VMware Player*** néven elkészített egy ingyenesen használható gépemulátort. Erről szól ***Szilágyi Attila*** cikke. ***Vécsei Balázs*** a ***Latex*** használatáról indít sorozatot, amire azonnal „rákontrázott” ***Kovács Levente***, aki a ***Kile*** nevű, „amit látsz azt kapod” típusú ***Latex*** szerkesztőt mutatja be. És ha már a szövegszerkesztésnél tartunk, ***Horváth Ernőtől*** megtudhatjuk, mire számíthatunk, ha dokumentumainkat különböző szerkesztők illetve különböző platformok között hordozzuk.

A közvélekedés szerint semmi jóra, ám ahogy ***Ernő*** cikkéből kiderül, a helyzet ennyire azért nem súlyos. Az alkalmazásokról szóló rovatot ***Fülöp Balázs*** cikke zárja, amely egy egyszerű de nagyszerű diagramszerkesztőt ismertet. Üzemeltetésről szóló rovatunkat ***Tóth Virgil Zoltán*** nyitja az ***IPCop*** bemutatásával. Olcsón vett laptopra ***Linuxot*** telepíteni sokak szerint „nagy kaland”. Hogy ez mennyire igaz (vagy nem), azt mutatja meg nekünk ***Tóth Péter***. Vannak persze manapság más kalandok is. Vehetünk például 64 bites rendszert akár otthonra, és kipróbálhatjuk, hogy futnak rajta régi, jól bevált alkalmazásaink. Ezt a kalandtúrát ***Medve Zoltán*** vezeti. ***Szabó Zoltántól*** az iskolai rendszerek üzemeltetésébe nyerünk újabb bepillantást, ***Sütő János*** pedig a ***clapf*** spamszűrőről írt, amit – nem mellesleg – ő maga fejleszt. A ***Grafika, multimédia*** rovatban folytatódik ***Lelovics Zoltán Scribus*** sorozata, de szó lesz egy digitalizálótábla telepítéséről, a fraktálokról, meg a száguldás örömeről is.

Kellemes időtöltést, jó szórakozást kíván a Linuxvilág stábja!

Hírek

Szundifigyelő

Ki ne utálná azt, amikor legszebb álmát szakítja meg az ébresztőóra csörgése? Így lehetnek ezzel az ***aXbo*** fejlesztői is, akik ráadásul azzal is tisztában vannak, hogy a mély álom nyújt igazán jó pihenést a szervezetnek. Akit eközben felkeltenek, az úgy fogja érezni, mintha semmit se aludt volna. Készítettek tehát egy csuklópántot, amely viselőjének alvás közbeni mozgásából képes következtetni az illető álmának mélységére, és a vele vezeték nélküli összeköttetésben álló ébresztőórával közli a keltés megfelelő időpontját. A tulajdonosnak csak azt kell megadnia, hogy mely időszakban kíván felkelni, ezt követően már a készülék határozza meg az ezen belül leginkább megfelelő pillanatot. A frissesség mindennapos érzése 50 ezer forintért költözik be a hálószobánkba.
➲ *www.axbo.com*

Végleges SeaMonkey

A ***SeaMonkey Council*** bejelentette a ***SeaMonkey*** internetes programcsomag 1.0-s változatát. A nyílt forrású, a ***Mozilla Application Suite*** alapjaira építkező csomag webböngészőt, levelezőprogramot, ***WYSIWYG*** stílusú weblapszerkesztőt és ***IRC***-ügyfélprogramot foglal magába. Természetesen az alkalmazások jellege, működése nem sok újdonságot tartogat azoknak, akik már ismerik a Mozilla alapítvány által a továbbiakban nem fejlesztett Mozilla ***Application Suite***-et, illetve a régebbi ***Netscape Communicatort***, az apróbb fejlesztések, javítások az ő számukra is tartogathatnak kellemes meglepetéseket.
➲ *www.mozilla.org/projects/seamonkey*

Minden helyzetre

Raptor 766X jelzéssel két ***VGA***-kártyával ellátott hordozható számítógépet mutatott be az ***Xtreme Notebooks.com***. Az ***Intel 915 GM*** lapkakészletre épülő gép eleve tartalmazza az ***Intel*** beépített grafikus vezérlőjét, amely visszafogott teljesítményével – mérsékelt fogyasztás mellett – kiválóan megfelel az irodai jellegű munkák elvégzéséhez. Amikor pedig a szövegszerkesztés és a táblázatgyártás véget ért, és kezdődik a játék, a másik vezérlő, egy ***PCI Express*** felületű ***nVIDIA 6600 MXM*** veszi át a vezérlést.
A 15,4"-os, szélesvásznú kijelzővel felszerelt gép a vásárló kívánsága szerint széles tartományban testreszabható, akár 2,26 GHz-es ***Pentium M*** processzorral és 2 GB memóriával is megrendelhető. További érdekessége a ***Silent Mode*** gomb, amellyel a beépített ventilátorok zaja mérsékelhető, amennyiben a környezet így kívánja.
➲ *www.xtremenotebooks.com*

DB2 a népnek

Az ***IBM*** elérhetővé tette a ***DB2 Universal Database Express Edition DB2 Express-C***, vagyis közösségi változatát. Az ***Express-C*** kiadás ingyenesen letölthető az ***IBM*** honlapjáról, és a fejlesztők arra is jogosultak, hogy saját termékeik mellé csomagolva továbbadják a végfelhasználóknak.
Az ***Express*** változat természetesen a nagyvállalati jellegű szolgáltatásokat nem biztosítja, azonban így is képes legfeljebb 2 db processzor és 4 GB memória kezelésére, ami belépő szinten bőségesen elegendőnek tűnik. Az ingyenes DB2 ***Linux*** és ***Windows*** alatt, 32 és 64 bites gépeken futtatható.

Multimédiás CyberLink-Linux

A főként ***DVD***-lejátszó alkalmazásáról ismert, de más termékeket is kínáló ***CyberLink*** bemutatta legújabb, ***PowerCinema Linux*** nevű termékét, amellyel, mint állítják, otthoni szórakoztató központtá lehet alakítani egy linuxos számítógépet. A ***PowerCinema Linux*** – kellő hardveres háttérrel – az analóg és digitális tévé- és rádióadások vételére egyaránt alkalmas, valamint kezeli a teletextes adásokat és az elektronikus műsorfüzeteket is. A mindössze 10 másodperces indítási és leállítási idejű program divatos, 3D-s kezelőfelülettel rendelkezik, ami a távirányítós vezérlési lehetőséggel párosítva a szórakoztatóelektronikai készülékekéhez teszi hasonlóvá a számítógép viselkedését. A széles körűen testreszabható csomagot jelenleg csak az eredeti készülékgyártók érhetik el.
➲ *www.cyberlink.com*

A fa örök

Nagy a világ, a technokrata ízlés mellett békésen megfér a konzervatívabb formatervezés is. Kiváló bizonyítéka ennek a holland *Guido Ooms* kínálata, amely faágakba épített *USB* kulcsokkal bővült. A kulcsok ránézésre, göcsörtösségük okán a mindennapi életben kevésbé tűnnek praktikusnak, ám ennyit – valamint 20 és 50 ezer forint közötti árat – igazán megér a szomszéd USB kulcsától való különbözőség.

Aki a praktikumot is szem előtt kívánja tartani, az inkább a Studio Leung kínálatából válogasson. A kemény fafajták mindig is minőséget sugalltak, nosztalgikus alkatok esetleg gyermekkori építőkockáikat is felidézhetik a szögletes fadarabokba épített USB-s adattárolók láttán.

➲ *www.oooms.nl*

➲ *www.studioleung.com*

Samba 4

Bemutatták a *Samba* csomag legújabb, 4-es számot viselő változatát – egyelőre előzetes jelleggel. A windowsos fájl- és nyomtatószolgáltatások helyettesítését célzó csomag *LDAP*-kiszolgálóval és *Kerberos* kulcskezeléssel bővülve immár az *Active Directory* kiváltására is alkalmas lesz, illetve képes lesz tartományi felhasználókezelési szolgáltatások biztosítására. Emellett további fejlesztések is történtek, a *Samba* csomagba beépítették a *SWAT* webes felügyeleti eszközt, az új parancsfájl-készítési felület révén Javascript alapú programok is kapcsolatba léphetnek a szolgáltatással, valamint új virtuális fájlrendszeri funkciók is megjelentek. A *Samba* fejlesztői hangsúlyozzák, hogy egyelőre előzetes, például a nyomtatást nem is támogató változatról van szó, amelynek termelési környezetben való használatát már csak a véglegesítésig várható változtatások nagysága miatt sem javasolják.

➲ *www.samba.org*

Drót nélkül, áram nélkül

Két új, *Power over Ethernet*-képes, vagyis a tápellátást az *Ethernet* hálózat kábelén fogadó *Wireless-G* hozzáférési pontot dobott piacra a *Linksys*. A *WAP54GP* jelzésű készüléket beltéri, míg a *WAP54GPE* jelű készüléket kültéri használatra ajánlják, illetve jellegükből fakadóan olyan helyekre, ahol a hozzáférési pont telepítésére ugyan szükség van, ám az áramellátás kiépítése túlságosan sok problémával járna. A két új hozzáférési pont támogatja a *WEP* és a *WPA* szabványt, továbbá ez év első negyedévének végére várható az a belsőprogram-frissítést, amelynek révén már a *WPA2*-t is ismerni fogják. A kényelemért sajnos meg kell fizetni, a kültéri *WAP54GPE* ára jelenleg százezer forint körül alakul.

➲ *www.linksys.com*

Összehúzták

Egy *SRAM* formájában elsőként az *Intelnek* sikerült teljes értékűen üzemelő, 45 nm-es eljárással készült lapkát előállítania. A következő generációs lapkák már ilyen eljárással készülnek majd – igaz, egyelőre a jelenlegi, 65 nm-es csíkszélességgel készülő termékek is újdonságnak számítanak az átlagfelhasználók körében, és maga a cég is csak két gyárát tudta átállítani erre a technológiára. A kisebb csíkszélesség a jelenlegi lapkákéhoz képest ötödakkora szivárgási áramot eredményez majd, így főként a mobil készülékek kedvezőbb fogyasztásában a kisméretű készülékek könnyebb működtetésében, no és persze a villanyszámlában jelent majd előnyt.

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 továbbiakban nincs szükség a *ksymoops* eszközre ahhoz, hogy a 2.6-os kernel alatt dekódolhassuk az *OOPS* üzeneteket. A visszafejtést immár maga a kernel is képes elvégezni, feltéve hogy fordításkor bekapcsoljuk a *CONFIG_KALLSYMS* opciót. Ez a kiegészítés lehetővé teszi, hogy ha az általunk futtatott rendszermag valamiért pánikszerűen leáll, a képernyőre kiírt adatokat egyszerűen bemásoljuk egy jelentésbe, és elküldjük azt a *linux-kernel* levelezési listára. Bár a kernelpánikkal kapcsolatban kényelemről beszélni talán némiképp költői túlzás, azért ez a szolgáltatás valamelyest megkönnyíti a tesztelők és egyéb szenvedő alanyok életét. A 2.4-es fa továbbra is a *ksymoops*-ra hagyatkozik, sőt teljesen valószínűtlen, hogy a fenti újítás bekerüljön ebben a változatba, hiszen annak pályafutása lassan a végéhez közeledik. *Marcello Tosatti* végül megnyerte az újításokkal szemben a 2.4-es mag stabilitásáért folytatott csatát, a jövőben pedig várhatóan még a jelenleginél is szigorúbban fogja elbírálni, hogy mi kerülhet be ebbe a fába.

Pantelis Antoniou megírta azt a kiegészítést, amelynek segítségével a szabványos *8250*-es soros kommunikációs meghajtón keresztül lehet kommunikálni az *AMD* által gyártott *Au1x00* beágyazott processzorral is. Ami azt illeti, különösebben nagy beavatkozásokra nem volt szükség, mivel a problémát csupán az jelentette, hogy az *Au1x00* regiszterei más eltolási címeken vannak, mint amire a *8250*-es meghajtó számított. Ezt az eltérést értelemszerűen egy megfelelő leképezőfüggvény segítségével lehetett áthidalni. Ezen kívül ki kellett iktatni a modem állapotváltozásaival kapcsolatos megszakításokat, mivel az *Au1x00* család nem minden tagja támogatja azokat. A szükséges részek egyelőre *#ifdef* blokkok formájában kerültek be a kódba, de a végleges alakot nyilván majd a fejlesztői közösség vitája fogja eldönteni. Bár az *#ifdef*-ek használata mostanság erős szemöldökráncolást vált ki a pedánsabb fejlesztők körében, az igazság az, hogy az ilyen szerkezetek használata a kernelforrásban egyelőre teljesen általánosnak nevezhető.

Jaya Kumar elkészítette az *AMD Geode CS5535* audioeszközöket támogató kódot, illetve fölvállalta a *CS5535*-ös *ALSA* meghajtó fejlesztésének koordinálását. Ami azt illeti, a dolog egészen könnyen ment. Szinte senkinek nem volt szakmai ellenvetése a kód kialakításával kapcsolatban. Még maga *Andrew Morton* is átnézte a forrást, de ő sem talált benne semmilyen komolyabb hibát. A fejlesztői csapat tervezi az *SPDIF* támogatás megvalósítását is, de *Jaya* máris jelezte, hogy neki nem áll módjában azt tesztelni. A *CS5535* gyakorlatilag egy a *GX 533@1.1W* processzorhoz és egyéb *Northbridge* összetevőkhöz tervezett kliensesz-köz, amit elsősorban beágyazott eszközökbe szánnak. Ami a rendszer *Linux* alatt történő használatát illeti, megvalósítható a fejlesztők eredeti elképzelése is, de az sem kizárt, hogy valami egészen briliáns csavarral valamiféle alternatív felhasználási mód honosodik majd meg linuxos körökben.

A kernel.org tartomány elsődleges kiszolgálója átköltözött az *Oregoni Állami Egyetem Nyílt Forrású Laboratóriumába (Oregon State University Open Source Lab)*, mivel itt nagyobb sávszélesség, nagyobb kiszolgálószemélyzet és jobb adatmentési lehetőségek állnak rendelkezésre. *Javier Henderson* saját kezűleg, egy magánrepülőgépen szállította el a gépet az új helyre, hogy ezáltal is csökkentse az állásidőt. Aki kíváncsi az eseményre, látogasson el a ➲ *http://osuosl.org/photos/kernel/view* címre és tekintse meg az „egy költözés képei” című fotókiállítást.

Az új helyen azért akadt néhány megoldandó probléma is. Gondok voltak a *kernel.org* tartomány más gépeire való tükrözéssel, illetve előjött egy olyan helyzet, amikor a különböző csomópontok a *git* tár különböző változatait hordozták, így a fejlesztők számos hibaüzenetet kaptak, amikor saját forráskód-állományukat a *Linus Torvalds* által fenntartott fához akarták szinkronizálni. Mindazonáltal a fenntartóknak mostanra sikerült kivasalni ezeket az apró gyűrődéseket és a *kernel.org* ismét tökéletesen működik.

A *Linux* bootolásért felelős kódja – mint számos más, ha nem az összes operációs rendszeré – meglehetősen összetett és kissé zavaros.

A közelmúltban ugyan volt egy elvetélt próbálkozás a rendbetételére, de ez csak annak felismerésére volt jó, hogy a dologhoz várhatóan sokkal több időre lesz szükség, mint azt eddig gondolták. ***Etienne Lorrain***, aki 1998 óta foglalkozik a bootkód fejlesztésével végül úgy határozott, hogy az eddig használt gubancos assembly kódot átírja szikrázóan tiszta *C*-re. „Apró mellékhatásként" ezt csak úgy tudta megtenni, ha felfüggeszti a ***LILO*** és ***GRUB*** rendszertöltők támogatását. Igaz ami igaz, ***Etienne*** számos kiváló kiegészítést készített az új kódhoz. Megszüntettet a kernel méretével kapcsolatos összes korlátozást, lehetővé tette a ***BIOS***-hoz való hozzáférést, de ahogy ***Pavel Machek*** rámutatott: ***„Volt nekünk egy csúnya assembly bootkódunk. Most van egy csodaszép C-ben megírt változat, ami viszont nem kompatibilis se a LILO-val, se a GRUB-bal. Akkor ez most mire is jó nekünk?"*** ***Stephen Hemminger*** a ➲ ***http://linux-net.osdl.org*** címen létrehozott egy új wikit a ***Linux*** hálózati dokumentációja számára. Itt bárki elhelyezheti a hálózatok kezelésével kapcsolatos dokumentációit. A lépés egyesekből, például ***Greg Kroah-Hartmanból*** némi ellenkezést váltott ki. ***Kroah-Hartman*** szerint a ➲ ***http://wiki.kernelnewbies.org*** címen régóta létezik egy ugyanilyen wiki, ahol Stephen nyugodtan elhelyezhette volna az általa fontosnak ítélt dolgokat. Természetesen ***Stephen*** válasza sem maradt el. Szerinte wikik eddig is voltak és ezután is lesznek. Ami meg a minőséget illeti, mindkét hely tartalma egészen jónak tűnik, és az sem kizárt, hogy egy napon majd egyesül a két tartalom.

Linux Journal 2006. 142. szám

Zack Brown

Web Konferencia 2006

Aki rendszeresen figyelemmel kíséri a hazai webes rendezvényeket, bizonyára izgatottan várja már a tavaszt, vele együtt a PHP Konferenciát. Hét nagy sikerű PHP rendezvényt követően a Neumann János Számítógép-tudományi Társaság Web Alkalmazások Fejlesztése Szakosztálya (NJSZT WFSZ) örömmel jelenti be új, Magyarországi Web Konferencia 2006 rendezvényét, amely március 18-án, szombaton kerül megrendezésre. Az esemény házigazdája ez alkalommal is ELTE Radnóti Miklós Gyakorlóiskola.

A névváltással fejezik ki a szervezők nyitottságukat a webes technológiák és megoldások szélesebb spektruma felé. Önmagában a lépés nem meglepő, hiszen az évek elteltével egyre több, nemcsak kizárólag *PHP*-s témával foglalkozó előadás hangzott el. Egyaránt teret kaptak a kliensoldali technikák és az általános webes kérdések is. A megújulást a rendezvény új külsőbe bújtatott honlapja és fejlődést, közösségi indíttatást tükröző logója is jelzi. A konferencia fő célja továbbra is a szakmai tapasztalatszerzés és a szakemberek, fejlesztői csoportok számára biztosított kötetlen fórum. Idén a *PHP*-fejlesztők mellett a ***Java***, ***.NET***, ***Ruby***, ***Perl*** és ***Python*** nyelv szerelmeseinek is nyújtanak izgalmas témákat a szervezők. A gazdag programot húsz előadás és az ebben a sorozatban újdonságnak számító a web jövőjéről folytatott kerekasztal-beszélgetés garantálja. Előtérbe kerülnek a web 2.0 megoldások, de idén is remélhetünk érdekességeket akadálymentesség és fejlesztési módszertan témakörökben. A ***Web Konferencia 2006*** szakmai színvonalát a sokéves tapasztalat, korábbi rendezvényekről ismert illetve professzionális szakemberek közül felkért új előadók és a látogatói visszajelzések teszik idén is felejthetetlenné. Ezúttal a rendezvény szakmai partnere lesz a ***W3C Magyar Iroda*** is.

A konferencián való részvétel ingyenes. Az elhangzott előadások anyagai a megszokott módon elérhetők lesznek a konferencia weblapjáról. Addig is a készülődésről és minden további részletről folyamatosan tájékozódni lehet a konferencia honlapján, mely a ➲ ***http://web.conf.hu/2006/*** címen található. A rendezvénysorozat ***RSS*** csatornája is elérhető a ➲ ***http://web.conf.hu/rss*** címen.

Várunk mindenkit sok szeretettel!

Kernel magyarul

Sokan nem mernek belevágni abba, hogy saját kernelt fordítsanak, mert nem beszélnek jól angolul. Igaz ugyan, hogy ez az indok kicsit instabil lábakon áll, hisz a számítástechnika nyelve mint tudjuk az angol. Akinek van valamelyes érzéke a digitális technikához, az a gép előtt ülve akarva-akaratlanul is elsajátítja a nyelv alapvető kifejezéseit, szavait. Ez így természetes, de mi a helyzet akkor, ha a felhasználónk nem akar semmi mást, csak használni a számítógépét. *„Akkor ne fordítson kernelt, mert nem neki való.”* – hallom a választ. De azért ne legyünk ilyen szigorúak.

Minden kezdet nehéz

A kernelkonfigurálásra egyrészt rá kell érezni és ismerni kell gépünk felépítését, másrészt meg a dolog „műveleti szinten” tényleg nem bonyolult: csak be kell jelölgetni a megfelelő opciókat. Természetesen a valóságban ez nem ennyire egyszerű, mert a megfelelő ismeretek hiányában kihagyunk valamit amitől nincs hang, hálózat vagy akár billentyűzet...
Szerencsére ezzel kapcsolatban született már rengeteg kiváló dokumentáció, amelyek lépésről-lépésre elmagyaráznak mindent. A figyelmetlenség ellen persze nincs védekezési módszer. Aztán lehet, hogy az előbb említett egyszerű felhasználó csak ki akar próbálni valamit, amihez viszont kell egy új, egyedi kernel. A változtatás lényege lehet a felesleges részek illetve modulok eltávolítása, de nem ritkán előfordul, hogy az általa vásárolt új hardvert (hangkártyát, infraportot és így tovább) nem támogatja az általa használt kernelváltozat. Aztán ha ilyenkor kérdezni mer egy fórumon, sajnos gyakran előfordul, hogy segítség gyanánt vagy a szótároldalt vagy egy nyelviskola linkjét kapja.

Ebben az esetben két lehetőségünk van. Nekiállhatunk szóról szóra lefordítani menüpontokat, ami meglehetősen hosszadalmas, ha beleszámítjuk, hogy számos opció csak a hozzá tartozó súgó alapján érthető meg. Kereshetünk azonban egy olyan dokumentumot is, amelyben megvan a fordítás, és némi magyarázat is. Nem is olyan régen megfogant egy ennél általánosabb megoldás ötlete is: honosítsuk a kernelkonfigurálás üzeneteit...

Történet

Az egész onnan indult, hogy körülbelül egy évvel ezelőtt a ***Linuxfórumon*** (➲ *www.linuxforum.hu*) felvetette valaki: miért ne lehetne honosítani a kernelkonfigurálást? Az ötlet – mint már megszokhattuk – heves indulatokat váltott ki. Voltak mellette és ellene egyaránt. Sok valóban megfontolandó ellenérv elhangzott (az informatika nyelve az angol, nehéz állandóan frissen tartani) néhányan mégis úgy döntöttünk, hogy belevágunk a dologba. Aztán lassan de biztosan érdektelenségbe fulladt a dolog, nem csatlakoztak hozzánk újabb emberek, de aztán egyszer csak változás történt: Jött ***Egry Gábor (gabaman)***, aki fantáziát látott a dologban. Ekkor alakult ki pontosan a cél és szerveződött meg a munka menete. A projekt hamarosan felkerült a ***SourceForge***-ra ***„The Linux Kernel Translation Project”*** néven (➲ *http://tlktp.sourceforge.net*) és ezzel együtt hamar nemzetközivé nőtte ki magát.
Az oldalon megtalálható minden fontos információ, többek között a különböző nemzetek jelenlegi fordítási aránya. Vannak, akik – köszönhetően a sok lelkes fordítónak – már el is készültek. Mi magyarok jelenleg 64%-on állunk, ami egészen jó arány ahhoz képest, hogy 3-4 embernél több nem dolgozott eddig.

A honosítás folt (patch) formájában érhető el a projekt honlapján. Ezt a kernelforrás könyvtárában kell kicsomagolni (ügyeljünk a verziószámok egyezésére) és megfoltozni vele a kernelt. Ezt követően, ha minden rendben ment, akkor a `make menuconfig` illetve `make xconfig` felülete már az anyanyelvünkön olvasható. A 64%-os állás azt jelenti, hogy a menüpontok teljesen magyarul vannak, viszont a súgószövegek egy része még angol.

Csatlakozz te is!

A fordítást önkéntesek végzik, akik szabadidejükből áldoznak arra, hogy segítsenek nekünk. Sajnos kevesen vagyunk, így minden segítő kéz számít. A feladat nem nehéz, csak le kell ülni és csinálni. Aki kicsit otthonosabban mozog az informatikai szaknyelvben, annak nem lesz nehéz, ezen kívül vannak nagyon jól használható on-line szótárak, illetve szótárprogramok. Fordításhoz a ***gtranslator*** vagy ***kbabel*** programokat ajánljuk, ezekben jól nyomon követhetőek a lefordított, fordítandó illetve ellenőrzésre váró szócikkek.
Ha van kedved tenni valamit a nyílt forrású világért, akkor látogass el a projekt oldalára és jelentkezz. Ha egyszer elkészül az egész fordítás, akkor már csak a frissebb kernelverziókban megjelenő újdonságokat kell fordítani azaz minimális munkával karbantartható lesz a nyelvi modul.

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.

Wiki és a tudásalapú társadalom

Valószínűleg sokunkkal előfordult már, hogy valamelyik kereső az egyik kulcsszóra egy wiki(pedia) oldallal tért vissza első találatként. Nos ez egyáltalán nem véletlen! Ha egy mondatban kéne összefoglalni a wiki-jelenséget, akkor azt mondanám, hogy az eddigi csak olvasható webtartalom helyett eljött az írható-olvasható oldalak kora.

Ebben a cikkben először bemutatom a *wikipedia.org*-ot, mint a legnépszerűbb wikit, aztán szólunk a wikik mögött megbújó szoftverekről és elmerengünk egy keveset a címben jelzett wiki-filozófiáról is. Talán ideje a ***Révai*** és a ***Britannica*** helyett valami másban gondolkodnunk?

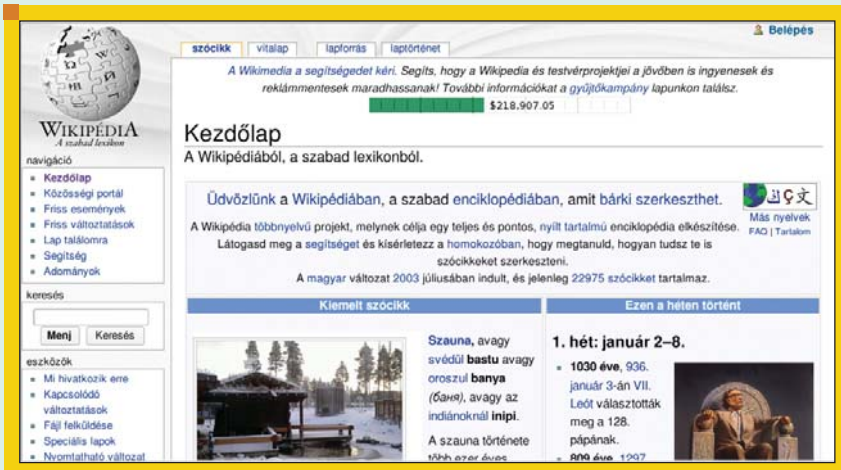
Wikipedia.org

Ha ellátogatunk a fenti címre, akkor először is nyelvet kell választszunk, hiszen a wikipedia többnyelvű alkalmazás. A magyar változat is egészen jól áll, már a több mint tízezer szócikket tartalmaz. Az oldal egyfelől használható lexikonként, hiszen ha beütünk egy kulcsszót, nagy valószínűséggel egy lexikon-oldalt kapunk vissza a témában. Nagy előnye, hogy a porszagú, több polcnyi lexikonokkal szemben itt nincsenek a fizikai méretek által behatárolt korlátok, a szócikkek szerzői legjobb tudásuk szerint tetszőleges terjedelemben kifejthetik a témát. A rendszer ezen kívül számtalan ötletes szolgáltatással igyekszik túllépni az egyszerű lexikonokon, hiszen idézetgyűjteményt, médiatárat, szótárat és fajbesorolót is találunk a testvérprojektek között. Ha pusztán a gyakorlati kezelhetőséget tartjuk szem előtt, akkor ötös alára vizsgázik a *Wikipedia*.

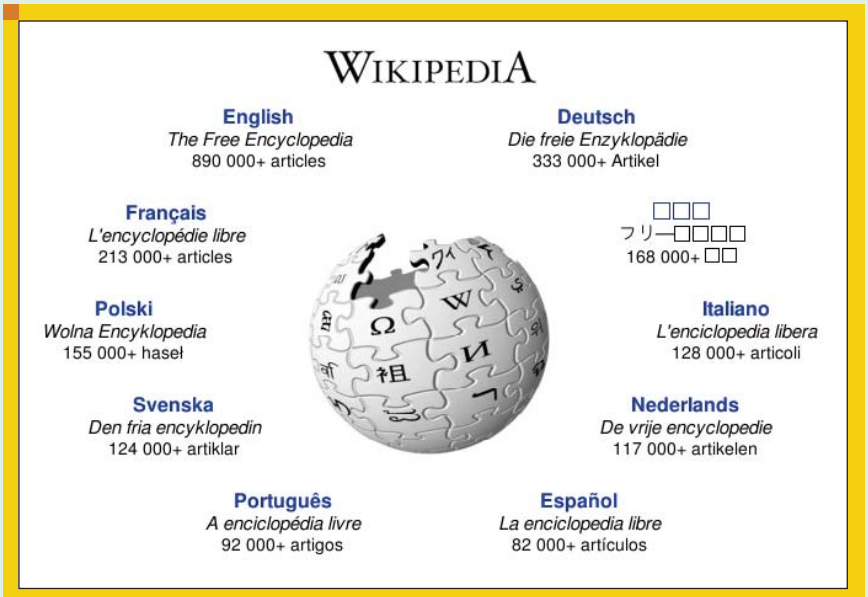
Magukat a szócikkeket nagyon könnyű megtalálni, ha nincs pontos találat, akkor a részleges egyezések listáját is megkapjuk, vagyis mindazon szócikkeket, ahol szerepel a keresett szó. Sok ábra és hanganyag is tartozik a lexikonhoz. Akkor hogyhogy ötös alá? A sebessége miatt. A legrosszabb időpontjaiban a *Wikipedia* vánszorog, a nagy mértékű felhasználói beavatkozások miatt nagyon lelassul. Persze ez valószínűleg csupán múló állapot: a kiszolgálópark erősödésével és esetlegesen a szoftverrendszer fejlesztésével megoldódhat.

A másik, ha lehet, még ennél is érdekesebb tevékenység a *wikipedia.org*-on az, ha az ember szócikkírásra adja a fejét. Ez általában úgy kezdődik, hogy felfedezünk valami apró hibát a lexikonban (ilyenek vannak, gyakran csak elgépelésből fakadnak) és azt kedvünk támad kijavítani. Semmi akadálya! Mint a bevezetőben is említettem, a Wiki írható és olvasható tartalom. Minden oldal tetején ott van a Szerkeszt/Edit gomb, mellyel egy szemvillanás alatt módosíthatjuk saját belátásunk szerint szinte bármelyik szócikket.

Ez persze rögtön felveti a virtuális vandalizmus kérdését, mely az egész ötletet fenyegeti. Erről majd a wiki-filozófia tárgyalásánál bővebben lesz szó. Első közelítésben semmi meg nem akadályozhat bennünket abban, hogy egy még nem létező témáról – ha elég naprakész a tudással rendelkezünk vele kapcsolatban – új szócikket alkossunk. Előbb azonban mindenképpen célszerű elolvasni az új szócikk létrehozásáról szóló szócikket. Itt megtudhatjuk, hogy mi való egy lexikonba és mi nem. Így alakul aztán a lexikon egyre nagyobb, egyre pontosabb ismereteket tartalmazó



1. ábra A wikipedia magyarul



2. ábra A wikipedia kezdőlapja

tudásbázissá. A szócikkeket szerencsés esetben olyan személy írja meg, aki alapos tudással rendelkezik a témáról. De ha nem, az se nagy baj, úgyis kijavítják.

Mediawiki

Aki hasznosnak találja a fent vázolt rendszert saját céljaihoz, az indíthat saját wikit. A wikipediát meghajtó szoftver nyílt forráskódú és *GPL* licensz alatt elérhető. Így aztán pár apró előkészület után könnyen közösségi portált tudunk létrehozni. Egy munkahelyen, vagy bármely olyan helyen, ahol több ember azonos célokon munkálkodik, hatékonyan meg lehet osztani információkat egy ilyen wiki segítségével.
Bizonyos esetekben akár egy levelezőlistát is kiválthat, mint a tudás központi forrása. Nyilván egy szűk felhasználói csoport esetén fontos korlátozni, hogy tényleg csak azok olvassák és írják az oldalt, akikre a dolog tartozik. Webes alkalmazásról lévén szó, talán nem meglepő, hogy a mediawiki beüzemeléséhez a szokásos „triumvirátus” szükséges: az Apache, a PHP és a MySQL.
Ez az összeállítás a legnépszerűbb a Linux alatt futtatott webes alkalmazások esetén, így a vállalati webkiszolgálón jó eséllyel pont ezeket találjuk majd.
Tekintsük át röviden az üzembe helyezés lépéseit.

Először töltsük le a ➲ *http://www.mediawiki.org/wiki/Download* címről magát a szoftvert. Csomagoljuk ki a *tar.gz* archívumot, aztán töltsük fel a szerverre a webszerver megfelelő könyvtárába (*htdocs* vagy *www*). A telepítővarázsló futtatásához adjuk ki a

```
chmod a+w config
```

parancsot, hogy a *PHP* szkriptek tudják írni az alkalmazás beállításait.
Ha ezután a a könyvtárra navigálunk egy böngészővel, a telepítővarázslót láthatjuk, ami már végigvezet bennünket a többi lépésen. A telepítés után mindösszesen annyi a teendőnk, hogy a *config* könyvtárat eltávolítjuk. Nincs más hátra, mint megtölteni tartalommal a létrejött rendszert, bár ez a legnehezebb feladat. A *wikipedia.org* a tartalma és nem a rajta futtatott szoftver miatt számít kuriózumnak. Jó ha tudjuk, hogy wiki nem csak egy van. Ma már számtalan ehhez hasonló írható-olvasható webes alkalmazás létezik. A *freshmeat.net* 140 alkalmazást talál a wiki kulcsszóra, ha tehát a fent említett mediawiki nem felel meg az igényeinknek, akkor a ➲ *http://freshmeat.net/search/?q=wiki* címen bízvást megtaláljuk az „igazit”.

A Wiki-filozófia

A *Wikipedia* rövid idő alatt nagy ismertségre tett szert. Ez elsősorban annak köszönhető, hogy egy ilyen vállalkozás láttán sok emberben felötlik egy hatalmas, közös, könnyen kezelhető tudás utópiája, mely nagyon vonzó alternatíva a Gutenberg-galaxishoz képest. Az angol kiadásnak jelenleg több mint háromszor annyi szócikke van, mint a nagy hírű magyar *Révai Nagy Lexikonnak* (896 ezer a 230 ezerrel szemben), így aztán kellő tekintélyt vívott ki magának ez a botcsinálta lexikonírók által összehordott ismerethalmaz. Talán az is a népszerűség jele, hogy már egy a wikipedia jelenséget ellenző weboldal is született: ➲ *http://wikipediaclassaction.org/*. Nyilván felmerül a kérdés, hogy mennyire hiteles a majd egymillió szócikk tartalma? A közelmúltban a magyar weben is megjelent több cikk amelyek ezt a kérdést feszegették?

3. ábra Egy telepített, testreszabott mediawiki

Nos, a jelek szerint a *Wikipedia* egészen jó! Egy független vizsgálat szerint az *Encyclopedia Britannicával* összehasonlítva szinte ugyanannyi az egy szócikkre eső tévedések száma. Ez valószínűleg annak is köszönhető, hogy egy-egy szócikk külön életet él. Valaki egyszer begépeli, aztán ezt sokan megnézik, átszerkesztik és sok szerkesztés után nyeri el többé-kevésbé végleges formáját. Persze végleges formáról ez esetben nincs is nagyon értelme beszélni, mivel gyakorlatilag bárki bármit átírhat.

Sajnos egyeseknek kiváló alkalom ez a digitális vandalizmus kiélésére is: nincs annak műszaki akadálya, hogy valaki egy teljes szócikket átírjon zagyvaságra. Az oldal üzemeltetői annyit tudnak tenni, hogy egy mozdulattal visszaállítják az előző állapotot (minden cikk teljes történetét tárolja a rendszer) és előbb-utóbb kitiltják az adott *IP* címet.

A rendszer felépítéséből adódik, hogy teljes védelmet nem nyújt a szándékos károkozással szemben, ezért aztán az angol *Wikipedia* korlátozásokat vezetett be: új szócikket nem írhatnak névtelen látogatók, de a meglévőeket továbbra is szerkeszthetik.

Vannak egész más elképzelések mentén szerveződő elképzelések is. Ilyen például a *The Digital Universe* (➲ *http://www.digitaluniverse.net/*) is, amely a *Wikipediához* hasonlóan egy tudásbázist szeretne létrehozni, de ezt nem akárki, hanem csak tudományos szakértők szerkeszthetik. A *Wikipedia* alapítói szerint ez nem jelent konkurenciát nekik, hiszen egészen másról van szó. Kicsit is értve és érezve a wiki-használatot, ez valóban így is van. Az egész dolog varázsát a szerkeszthetőség adja, az idő pedig majd eldönti, hogy melyik elképzelés végzi a net süllyesztőjében.

Addig is használjuk az internet e roppant méretű, rendszerezett információhalmazát, írjunk bele és javítgassuk. Akár létrehozhatjuk a magyar *Wikipediában* a *Linuxvilág* szócikket is. Csak arra ügyeljünk, hogy a szerkesztés során legyünk pártatlanok és kiegyensúlyozottak. Nem ajánlják, hogy saját magunkról illetve velünk szoros kapcsolatban lévő dologról írjunk cikket. Így aztán valamelyik olvasóra vár ez a nemes feladat.

A *Linux* és egy kreatív ötlet segítségével akármikor a *Nagy Francia Enciklopédia* híres szerkesztőinek bőrébe bújhatunk egy időre, hiszen ahogyan az enciklopédia ötlete és szellemisége merőben új volt akkor, éppúgy csodálkoznak ma a *Wikipedián*.

Novák Áron
(aaron@szentimre.hu)

BME-VIK-es hallgató, műkedvelő rendszergazda. Jelenleg leginkább a NetBeans-szel és mindenféle hordozható eszközzel foglalkozik, legalábbis mindazokkal, amelyeket meg lehet szólaltatni Linux alatt.

KAPCSOLÓDÓ CÍMEK

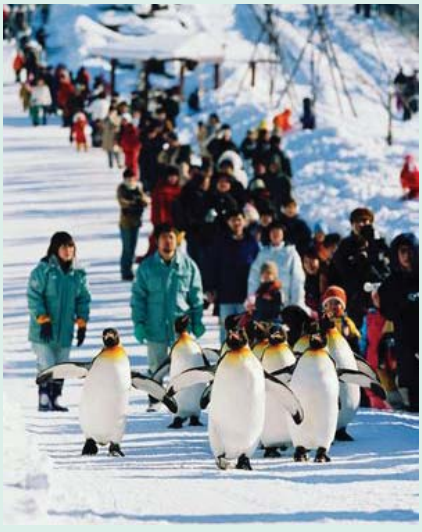
Magyar wikipedia:
➲ http://hu.wikipedia.org

Angol wikipedia:
➲ http://en.wikipedia.org

Fogyasszunk pingvint!

Mi magyarok madárszékleteket tanulmányozunk nyomás alatt, az amerikaiak Nobel-díjakat osztogatnak papírrepülőkkel, a franciák a Déli-sarkon filmforgatva szentektől s melegektől zsebelnek be dollár tízmilliókat, a norvégok hadereje plecsniket osztogat az állatkertben – a japánok pedig pingvint fogyasztanak. No nem késsel-villával...

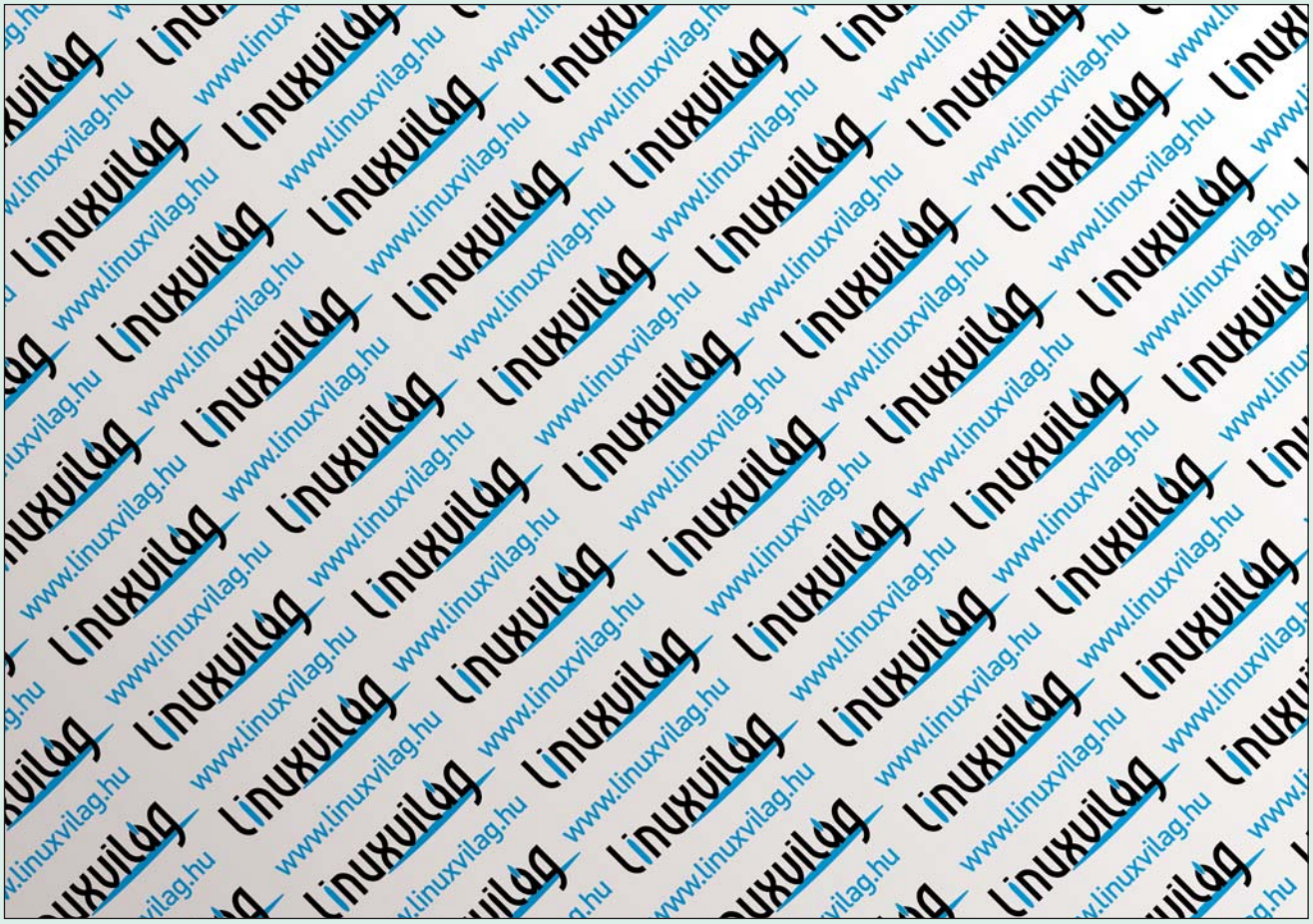
Úgy tűnik, az állatvilágot sem kerüli el a szezonális hízás, s tél végén szembesülve a mozgásszegény életmód okozta testtömeg-többlettel, intenzív alakformálásba kezdenek. Noha pingvinéknél talán beválik az a soványító módszer, hogy ha karcsúbbaknak akarunk látszani, vegyük körül magunkat kövérekkel, a hokkaidói ***Asahiyama*** állatkert gondozói mégis úgy vélekednek, eredendő testalkat ide, hideg elleni hájnövesztés oda: madaraiknál már igazán túlzott ez a téli túlsúly.
Csak lassan a testtel? Mivel pingvin formával korlátozottak a sportolási lehetőségek – a bicikli nyergéből nem ér le a lábuk, a focilabdához nem férnek a hasuktól, a kosárlabdához alacsonyak, a szumóhoz vékonyak, a teniszhez zömökek, a kungfuhoz lassúak és még sorolhatnánk –, így a legideálisabb társas sportot választották számukra:



a kocogást. Talpára és libasorba állítva a japán pingvincsapatot, naponta kétszer 500 métert masíroznak erőltetett tempóban egy hóval felszórt kifutón, s ez így megy majd napról-napra, míg nem a tavaszi langyidő beköszöntével elköszönhetnek fölös tömegüktől.
Hogy ne érje külalaki diszkrimináció a madárvilágot, említtessék meg néhány hasonló állati karcsúsodási törekvés. Itt van mindjárt ***Maggie***, a depressziós alaszkai elefánt, ki bánatában felszedett egy-két mázsácskát. S mivelhogy ő sem az a kimondott siklóernyős-vízisíző típus, személyi edzője őt is kocogásra bírta. Hol és miként lehet egy elefánttal futkározni? Nos, 6 méter hosszú és cirka 7 tonna súlyú futópadot építve.
Hogy lássuk, a kocogás mily jótékony hatással bír, a migrénnel, fáradékonysággal és pajzsmirigy-problémákkal küzdő chicagói orángutánleány, ki magába (és 50 pluszkilójába) roskadva már leányságáról is megfeledkezett, a diétának és kitartó edzésnek köszönhetően immáron az orángutánlegényeket kergeti.
Azt mondják, ha kövér a kutyád, keveset mozogsz. Nemzetközi felmérés szerint a világ minden negyedik kutyáján és macskáján lassan már bányászeszközökkel lehet csak a bordákat kitapintani – e hírre a liverpooli egyetem máris magánklinikával specializálódott testes házikedvencek leapasztására. Valószínűleg a klinika szolgáltatásai elkerülték figyelmét a berlini ***Mikeschnek*** és a minnesotai ***O.T.***-nek, mely nevek alatt és vagdalthúsok felett dagadó kandúrok túlhaladták a 18 kilogrammot. Talán nem is csoda, ha a vagdalthús mellett alternatívaként a washingtoni kutatók kísérlet-egere marad, kin bizonyítva, hogy a szervezetben bizonyos bélrendszeri baktériumtörzsek megléte vagy hiánya alapvetően befolyásolhatja a derékbőséget, a kísérleti alanyt duplájára-triplájára hizlalták.
Bíztató az érintetteknek, hogy már „kövérségbaktériummal" is menteget-hető a chips-evés és (sport gyanánt)

a távolbanézés, és egyre-másra fejlesztgetik a fogyasztó gyógyszereket; emellett úgy látszik, eljött az idő, mikor az állatok kezdenek diétába, hogy a fogyasztónak ne kelljen...

Halusz Léna



Programfejlesztés az OpenGL segítségével (2. rész)

Irrlicht 3D

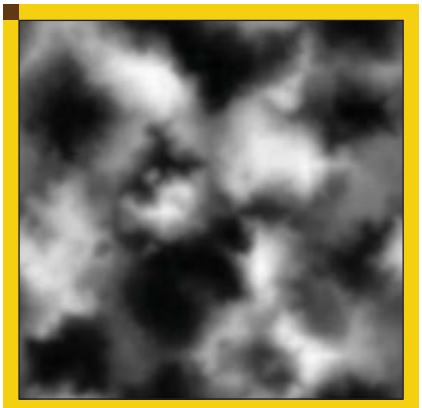
Az előző OpenGL cikkben megismerkedtünk a GLUT-tal és az OpenGL alapjaival. Most egy kis kitérőt teszünk a játékfejlesztés felé, megismerkedünk egy kiváló, nyílt forráskódú 3D engine-nel az Irrlicht-tel és, hogy saját OpenGL-es tapasztalatunkat is tovább bővítsük betekintünk az OpenGL anyag és fény kezelésébe is.

Mi az a 3D motor?

Sokan vagyunk akik már játszottak életükben legalább egy *3D*-s játékkal, de igen kevesen vannak azok akik bele gondoltak már, hogy micsoda munka elkészíteni egy ilyen játékot. A három dimenziós játékok „szíve” egy motor amely rengeteg munkát végez: kezdve az objektumok és a rajtuk található mintázatok megjelenítésétől a fizika modellezésén keresztül a speciális effektek megjelenítéséig. Általában ezek a motorok több jól elkülöníthető részből állnak, hiszen ostobaság volna mindent egybe zsúfolni.
Egy játék motor általában az alábbi részekre bontható szét:

- Grafikai megjelenítés
- Fizika
- Hangrendszer
- Mesterséges intelligencia
- Hálózatkezelés

Ezek a részek természetesen tovább bonthatók további részfeladatokra is, de mi most csak a grafikával fogunk foglalkozni.

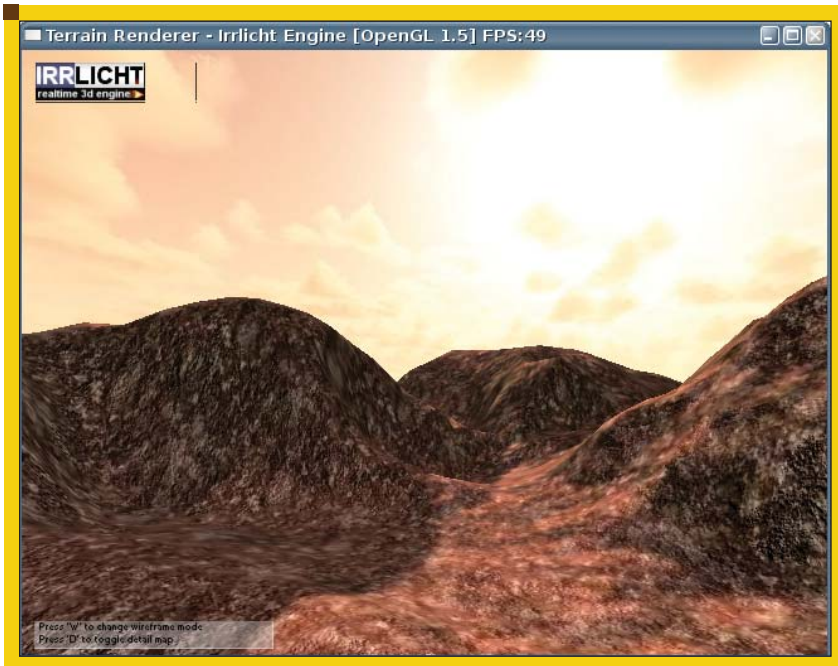


1. ábra Egy felülnézeti domborzati kép...

2. ábra A felülethez tartozó textúra

A megjelenítés

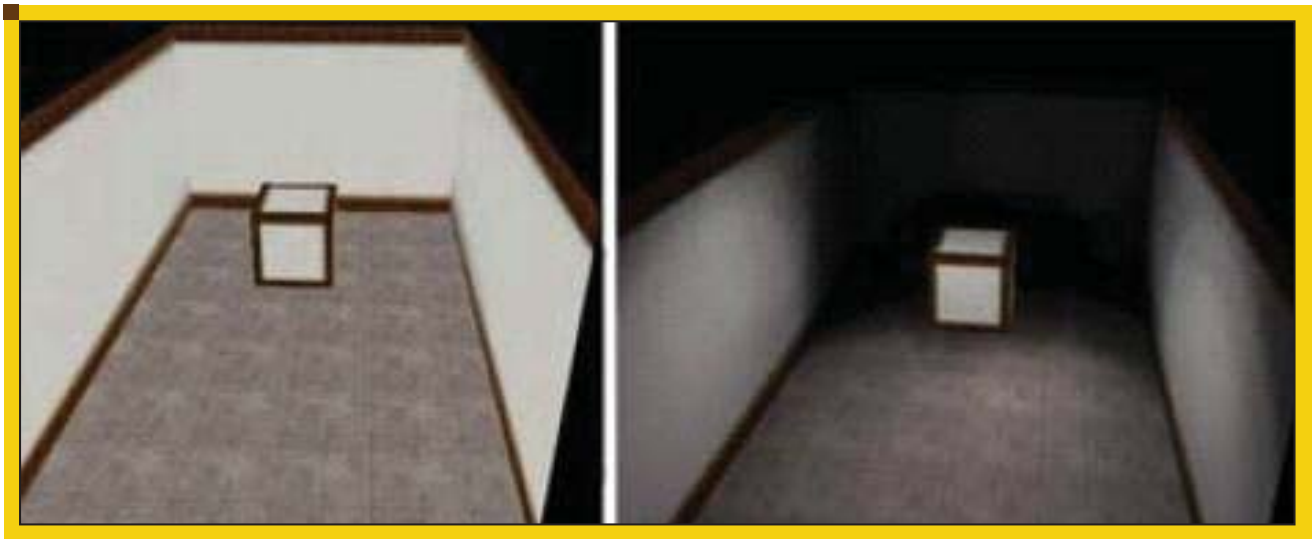
A legelső szempont amit egy grafikai rendszer megtervezésénél figyelembe kell venni, hogy milyen feladatot fog betölteni az általunk fejlesztett alkalmazás. Ha *PacMan*-t szeretnénk írni akkor teljesen felesleges *3D*-s objektumokkal foglalkozni. A „piacon” lévő engine -ek (*Source*, *Quake 1-2-3*, *LithTech*) nagy része *FPS* nézetre hajaz, de találhatunk olyat is ami általános grafikus motorként is megállja a helyét, így bármilyen típusú alkalmazásban megállja a helyét.
Sok helyről hallottam, hogy: *„Én az OpenGL/DirecX engine-t használom”*. Sem az *OpenGL*, sem a *Direct3D* nem grafikus motor. Ezek alacsony szintű programozói könyvtárak, amelyekre a bátor titánok kellő bátorság és szakértelem birtokában építhetnek egy motort. Hogy mi a különbség? Az *OpenGL*-t és a *D3D*-t azért nevezzük alacsony szintű *API*-nak,



3. ábra ...és a végeredmény!

mert a tudásuk csak az alapvető három dimenziós műveletek megvalósításáig terjed, míg egy valódi enginek ennél sokkal többet tudnak. Nézzük meg, hogy mik ezek az „extra szolgáltatások”!
A legalapvetőbb igény ami egy grafikai rendszerrel kapcsolatban felmerülhet az, hogy kezeljen objektumokat és ezeket a megfelelő adatok alapján el tudja helyezni a térben. A modellek típusától függően több fájltípus támogatása az optimális.
Animált objektumok (játékosok, szörnyek, …) legegyszerűbben *MD3* fájlformátumban tárolhatók.

Ezt a fájlszerkezetet az *idSoftware* fejlesztette ki a *Quake* játékokhoz. Az *MD* család lényege, hogy képes „csontváz” felépítésű modellek kezelésére. Ez azt jeleni, hogy a modellezők egy valódi csontrendszer elkészítése után rá tudják „húzni” a testet a csontvázra és a továbbiakban nem kell minden egyes animációs fázisban az összes testrész mozgását megszerkeszteni, elég csak a csontvázat mozgatni amely a ráhúzott testet a megfelelően módosítja. Egy ilyen fájlban el vannak tárolva különböző mozdulatsorok amelyekre az adott karakter képes lehet (gyaloglás, támadás, ugrás, esés, …). Az engine-nek képesnek kell lennie a fájl betöltésére és a mozgási fázisok animált megjelenítésére.
Statikus objektumok (asztal, szék, …) ennél lényegesen egyszerűbbek hiszen nem kell őket animálni, elégséges csak egyszer megszerkeszteni őket. Erre a feladatra kiváló a *3DS*, *OBJ*, *LWO*, *X* (és még sorolhatnánk) formátumok bármelyike.
Nagyon fontos, hogy a modellezők által elkészített objektum hiteles legyen ezért nem szabad megfeledkeznünk a textúrázásról sem. A fent említett fájltípusok mindegyike képes tárolni textúrázásra vonatkozó adatokat.
Maradt még egy harmadik objektum típus is: a pályaszerkezet. Ez nem más mint az objektumokat körülvevő világ. Beláthatjuk, hogy felesleges volna egy hegységrendszer minden egyes pontjának az előzetes eltárolása ezért itt alternatív megoldások után kell néznünk. Az egyik legegyszerűbb eljárás a domborzati kép eltárolása. Egy két dimenziós képen eltároljuk a domborzatot úgy, hogy csak fekete és fehér színeket, illetve ezek átmenetei használjuk.
Ezzel a módszerrel már lehet szép tájakat készíteni, de még mindig nincsenek belső tereink, épületeink. Erre szintén az *idSoftware* által kidolgozott fájlformátum kínálja az egyik legjobb alternatívát ami a *BSP*. A *BSP* fájlformátum lényege, hogy tárolja a tér geometriai felépítésén kívül a hozzá tartozó *BSP* fát és az előre számolt megvilágítási modelleket is.



4. ábra Radiosity-vel és nélküle...

1. táblázat

Paraméter neve	Alapérték	Leírás
GL_AMBIENT	(0.0, 0.0, 0.0, 1.0)	A fény ambient RGBA erőssége
GL_DIFFUSE	(1.0, 1.0, 1.0, 1.0)	A fény diffuse RGBA erőssége
GL_SPECULAR	(1.0, 1.0, 1.0, 1.0)	A fény specular RGBA erőssége
GL_POSITION	(0.0, 0.0, 1.0, 0.0)	A fény (x, y, z, w) pozíciója a térbe
GL_SPOT_DIRECTION	(0.0, 0.0, -1.0)	A fény (x, y, z) iránya (irányvektor), csak reflektor típusnál adjuk meg! Ha elhagyjuk akkor pont típusként fog viselkedni.
GL_SPOT_CUTOFF	180.0	Reflektorfény sugárzásának kúpszöge.
GL_CONSTANT_ATTENUATION	1.0	Konstans tompítófaktor
GL_LINEAR_ATTENUATION	0.0	Lineáris tompítófaktor
GL_QUADRATIC_ATTENUATION	0.0	Négyzetes tompítófaktor

A ***BSP*** fa lényege, hogy megkímélje a processzorunkat a felesleges munkától úgy, hogy eltárolja, hogy melyik pontból melyik pontokat ***nem*** látjuk. Ezáltal nem kell kirajzolnunk olyan dolgokat amelyek nem is esnek bele a látóterünkbe. Akit mélyebben érdekel a BSP működése az nézzen utána a ➲ ***www.gamedev.net*** oldalon a ***Binary Space Partitioning (Bináris Térfelosztás)*** című témának.
Természetesen a pályákat tárolhatjuk ugyanolyan statikus formátumban is mint az objektumainkat, de akkor nekünk kell kiszámolnunk, hogy melyek azok a felületek amelyen nem láthatók és mindemellett olyan extráktól is elesünk mint a ***radiosity*** ami nem más mint a fény előre kiszámított visszaverődés a különböző objektumokról.
Ez olyasmi mint a sugárkövetés, de itt a pályát leíró fájlstruktúrában tároljuk el a fény útját ezért lényegesen szebb képet kapunk mintha nem használnánk ***radiosity***-t, viszont meg sem közelíti egy valós sugárkövetéssel készült kép minőségét. A ***radiosity*** algoritmus csak statikus fényekre vonatkozik a ***BSP***-ben! Értelemszerűen a futásidőben létrehozott fények (rakéta, tűzgolyó, …) számára nekünk kell számolnunk, vagy más alternatíva után néznünk. Ahhoz, hogy virtuális világunk még tökéletesebb legyen nem szabad elfelejtenünk a fényeket sem.
A fényeknél megvilágítási modelleket definiálhatunk amelyek leírják a fények és a modelltér kapcsolatát. Az ***OpenGL*** által támogatott három megvilágítási modell: szórt háttér világítás (***ambient***), diffúz visszaverődés (***diffuse***) és a fényvisszaverődés csillogó felületekről (***specular light***). Ha egy fényt hozunk létre ***OpenGL*** segítségével akkor a fenti három tulajdonságot mindenképpen meg kell adnunk, illetve meg kell határoznunk, hogy a fényforrásunk pont (nap), vagy reflektor típusú-e (elemlámpa). A három fő tulajdonság (***ambient, diffuse, specular***) beállítása az RGBA paletta használatával történik a már jól ismert ***OpenGL***-es módon. Minden érték 0-tól 1-ig vehet fel értéket. Ha ezeket beállítottuk akkor jöhetnek az egyéb beállítások is (*1. táblázat*). Az utolsó három tulajdonság a fény gyengülését határozza forrástól való távolság függvényében.

A fény tulajdonságainak beállítása a `glLightfv` paranccsal történik a következő módon:

```
float light_ambient[]  = { 0.0,
➥0.0, 0.0, 1.0 };
float light_diffuse[]  = { 1.0,
➥1.0, 1.0, 1.0 };
float light_specular[] = { 1.0,
➥1.0, 1.0, 1.0 };
float light_position[] = { 1.0,
➥1.0, 1.0, 0.0 };
float light_direction[] = {
➥12.0, 14.0, -3.0 };
float light_cutoff[] = { 90.0 };

glLightfv(GL_LIGHT0,
➥GL_AMBIENT, light_ambient);
glLightfv(GL_LIGHT0,
➥GL_DIFFUSE, light_diffuse);
glLightfv(GL_LIGHT0,
➥GL_SPECULAR, light_specular);
glLightfv(GL_LIGHT0,
➥GL_POSITION, light_position);
glLightfv(GL_LIGHT0,
GL_SPOT_DIRECTION,
➥light_direction);
glLightfv(GL_LIGHT0,
➥GL_SPOT_CUTOFF, light_cutoff);
```

2. táblázat

Paraméter neve	Alapérték	Leírás
GL_LIGHT_MODEL_AMBIENT	(0.2,0.2,0.2,1.0)	A szórt háttérvilágítás intenzitása (RGBA)
GL_LIGHT_MODEL_LOCAL_VIEWER	0.0	Hogyan számolódjon a fény specular fényvisszaverődés szöge.
GL_LIGHT_MODEL_TWO_SIDE	0.0	Egy vagy két oldalas megvilágítás. Ha kétoldalas ra állítjuk akkor a felület belső oldalának anyagjellemzői is beleszámítanak a fényvisszaverődésbe.

3. táblázat

Paraméter neve	Alapérték	Leírás
GL_AMBIENT	(0.2, 0.2, 0.2, 1.0)	Az anyag ambient RGBA tükröződése
GL_DIFFUSE	(0.8, 0.8, 0.8, 1.0)	Az anyag diffuse RGBA tükröződése
GL_SPECULAR	(0.0, 0.0, 0.0, 1.0)	Az anyag specular RGBA tükröződése
GL_EMISSION	(0.0, 0.0, 0.0, 1.0)	Az anyag saját fénye
GL_SHININESS	0	A fényvisszaverődéskor létrejövő „fényfolt” intenzitása. Minél magasabb annál élesebb.

1. Lista

```
void myinit(void)
{
//A fény tulajdonságainak beállítsa (ambient,
diffuse, elhelyezkedés
  GLfloat ambient[] =  {0.0, 0.0, 0.0, 1.0};
  GLfloat diffuse[] =  {1.0, 1.0, 1.0, 1.0};
  GLfloat position[] = {0.0, 3.0, 3.0, 0.0};

  GLfloat lmodel_ambient[] =  {0.2, 0.2, 0.2,
  ➥1.0};
  GLfloat local_view[] =  {0.0};

//A fény létrehozása a tulajdonságok
//megadásával...
  glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);
  glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse);
  glLightfv(GL_LIGHT0, GL_POSITION, position);
//..és a megvilágítási modell definiálása
//(alap beállítások)
  glLightModelfv(GL_LIGHT_MODEL_AMBIENT,
  ➥lmodel_ambient);
  glLightModelfv(GL_LIGHT_MODEL_LOCAL_VIEWER,
  ➥local_view);

//A fények engedélyezése
  glFrontFace(GL_CW);
  glEnable(GL_LIGHTING);
  glEnable(GL_LIGHT0);
  glEnable(GL_AUTO_NORMAL);
  glEnable(GL_NORMALIZE);
  glEnable(GL_DEPTH_TEST);
  glDepthFunc(GL_LESS);
}
void renderTeapot(GLfloat x, GLfloat y,
  GLfloat ambr, GLfloat ambg, GLfloat ambb,
  GLfloat difr, GLfloat difg, GLfloat difb,
  GLfloat specr, GLfloat specg, GLfloat specb,
  ➥GLfloat shine)
{
  float mat[4];
  glPushMatrix();
  glTranslatef(x, y, 0.0);

//Ambient összetevő beállítása
  mat[0] = ambr;  mat[1] = ambg;  mat[2] =
  ➥ambb;  mat[3] = 1.0;
  glMaterialfv(GL_FRONT, GL_AMBIENT, mat);

//Diffuse összetevő beállítása
  mat[0] = difr;  mat[1] = difg;  mat[2]
  ➥= difb;
  glMaterialfv(GL_FRONT, GL_DIFFUSE, mat);

//Specular összetevő beállítása
  mat[0] = specr;  mat[1] = specg;  mat[2]
  ➥= specb;
  glMaterialfv(GL_FRONT, GL_SPECULAR, mat);

  glMaterialf(GL_FRONT, GL_SHININESS, shine
  ➥* 128.0);
  glutSolidTeapot(1.0);
  glPopMatrix();
}
```

És végül meghívhatjuk a renderTeapot függvényt megfelelően paraméterezve:

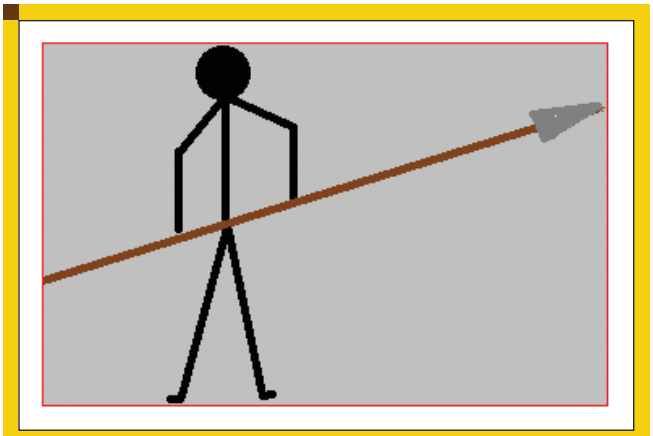
```
//Teáskanna smaragdból...
  renderTeapot(2.0, 17.0, 0.0215, 0.1745,
  ➥0.0215,
    0.07568, 0.61424, 0.07568, 0.633, 0.727811,
    ➥0.633, 0.6);
//...ezüstből...
  renderTeapot(6.0, 2.0, 0.19225, 0.19225,
  ➥0.19225,
    0.50754, 0.50754, 0.50754, 0.508273,
    ➥0.508273, 0.508273, 0.4);

//...és fekete gumi kiszerelésben.
  renderTeapot(14.0, 17.0, 0.02, 0.02, 0.02,
  ➥0.01, 0.01, 0.01, 0.4, 0.4, 0.4, .078125);
```

5. ábra A három teáskanna

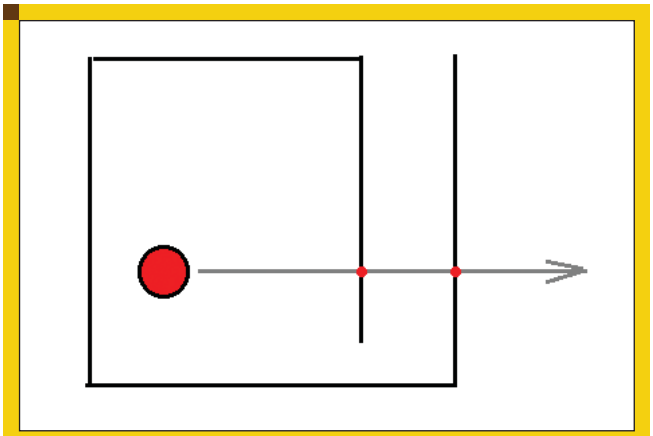
Jól látható, hogy első paraméter a fény azonosítója `GL_LIGHT7`-ig, majd ezt követi a változtatandó tulajdonság konstans neve és végül pedig a tulajdonság leírása vektor formájában.
Ha már létrehoztunk egy fényforrást, akkor azt is be kell állítanunk, hogy milyen megvilágítási modellt kívánunk használni. Ez a `glLightModelfv` utasítással történik amelynek paraméterként megadhatjuk, hogy egy-vagy két oldali megvilágítás modellt kívánunk-e használni, illetve itt definiálhatjuk a modelltér szórt háttérvilágításának intenzitását is (*2. táblázat*).
Már meg tudjuk világítani a teret, de még valami mindig hiányzik: az anyagjellemzők. Ezek azért fontosak, mert egy vas golyóról másképpen verődik vissza a fény mint egy narancsról, vagy mint egy üveggolyóról.
Az anyagjellemzők definiálása hasonlóan történik mint a fény tulajdonságainak beállítása.
Amikor egy objektum anyagát definiáljuk, nem azt mondjuk, hogy ez a teáskanna piros legyen, hanem úgy állítjuk be az anyagjellemzőket, hogy a piros fényt verje leginkább vissza. Ez a valóságban is így működik.
A fekete azért fekete, mert a ráeső fény nagy részét elnyeli, míg a fehér mindent visszaver. Az ***OpenGL*** segítségével meg kell határoznunk, hogy melyik fény típusokból (***ambient, diffuse, specular***) mennyit ver vissza az objektum.
Az anyag tulajdonságait a `glMaterialfv` paranccsal tudjuk beállítani, amelynek paramétereit a *3. táblázatban* láthatjuk.
Az *1. Listában* látható egy példa, amit az ***OpenGL Redbookból*** emeltem ki. Ebben a példában igen jól látható, hogy milyen egyszerű fényforrásokat, megvilágítási modellt és anyagjellemzőket beállítani az ***OpenGL*** segítségével.
A mostani cikkben ennyi az ***OpenGL*** specifikus rész, úgyhogy most kanyarodjunk is vissza eredeti témánkhoz a ***3D*** motorokhoz.
Most már tudjuk hogyan épülnek fel az alapvető modellek és pályák, valamint tisztában vagyunk a megvilágítás és az anyagjellemzők kapcsolatával is. Ejtsünk most egy pár szót a fizikáról, azon belül is programozástechnikailag az egyik legbonyolultabb alapvető műveletről az ütközésvizsgálatról.

Az ütközésvizsgálat (Collision detection)

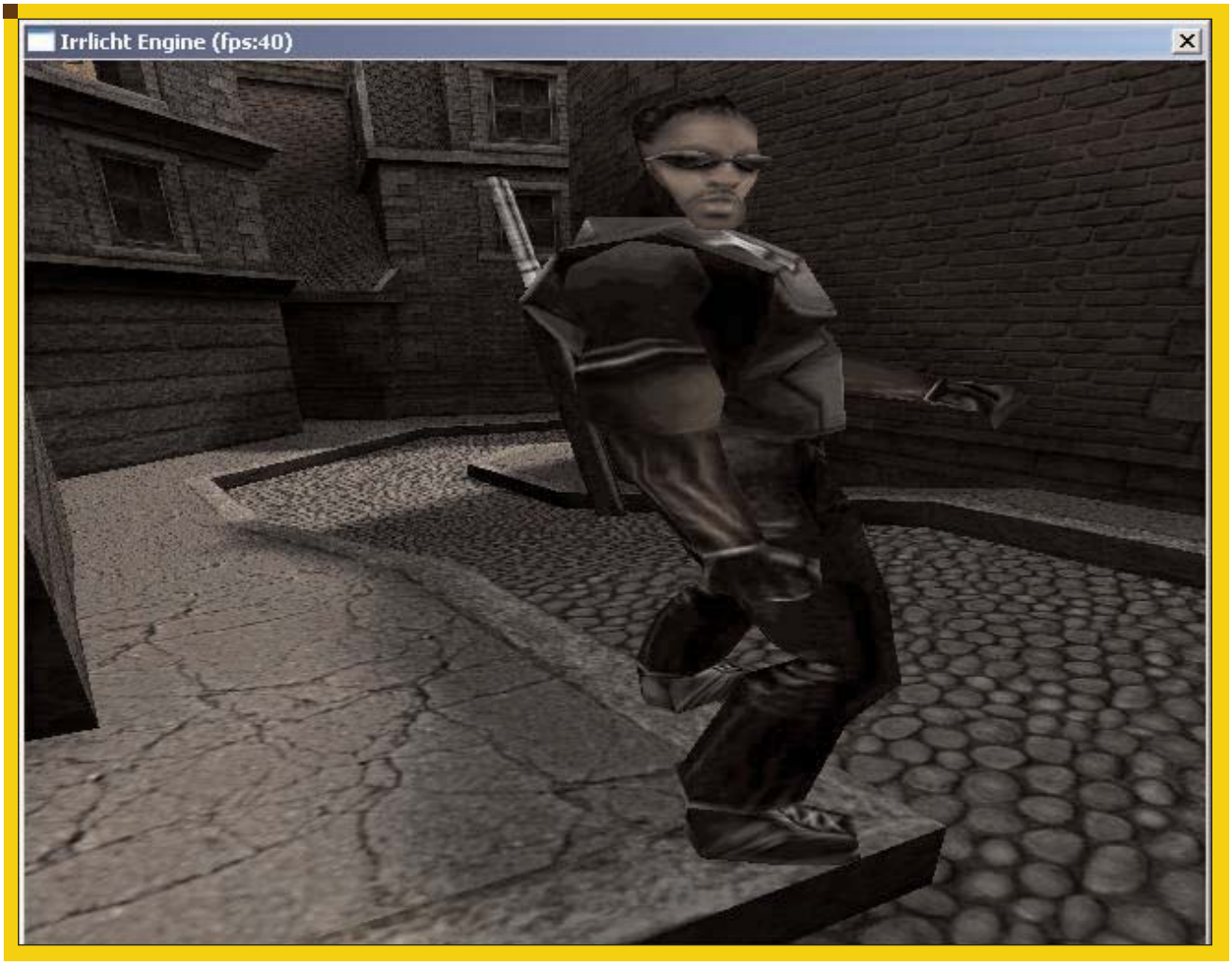
Ütközésvizsgálat nélkül a legtöbb játék játszhatatlan lenne, ezért igen nagy hangsúlyt kell fektetni a programozóknak egy gyors és pontos algoritmus kifejlesztésére. Az ütközésvizsgálat egy viszonylag nagy számítás igényű eljárás ezért célszerű a lehető legjobban optimalizálni ezt az algoritmus. Hogyan döntjük el, hogy két objektum metszi-e egymást? A legegyszerűbb módszer ha a játékban lévő minden objektum koordinátáját (pontosabban azoknak minden egyes „face”-ét) összehasonlítjuk az összes többivel. Egy 200 ezer poligonból álló objektum rendszer esetén ez „200000 a négyzeten” lépésből állna. Egyes játékok több millió poligonnal dolgoznak, úgyhogy ezt az eljárást el is vethetjük. A másik igen primitív eljárás, hogy a kamera látóterébe eső objektumokkal foglalkozunk, így jelentősen csökkenthető ennek az algoritmusnak a műveletigénye. De sajnos ez sem az igazi.
Másik igen butácska algoritmus a „dobozolás” módszere. Ezt már jó hatásfokkal lehet ***2D***-s játékoknál, vagy ***izometrikus (Diablo, Starcraft, Fallout)*** felépítésűeknél alkalmazni, de ***3D***-ben kerülendő a használata!
A lényege, hogy minden test köré teszünk egy akkora téglatestet amibe pontosan belefér, és ezeknek a téglatesteknek az ütközését vizsgáljuk. Egy 20-30 ezer poligonból álló ***MD3*** modellnél így már csak 12 poligonnal kell számolnunk. Ez igen jelentős sebesség javulást eredményez, viszont vannak olyan objektumok amiket nem lehet így bezárni, vagy ostobaság lenne ezt tenni.
Az alábbi képen egy stilizált ember figura van dárdával a kezében (ha valaki nem ismerte volna fel).



6. ábra Feleslegesen nagy „doboz” a lándzsás harcos körül



7. ábra A mozgó test irányvektora két poligont metsz el

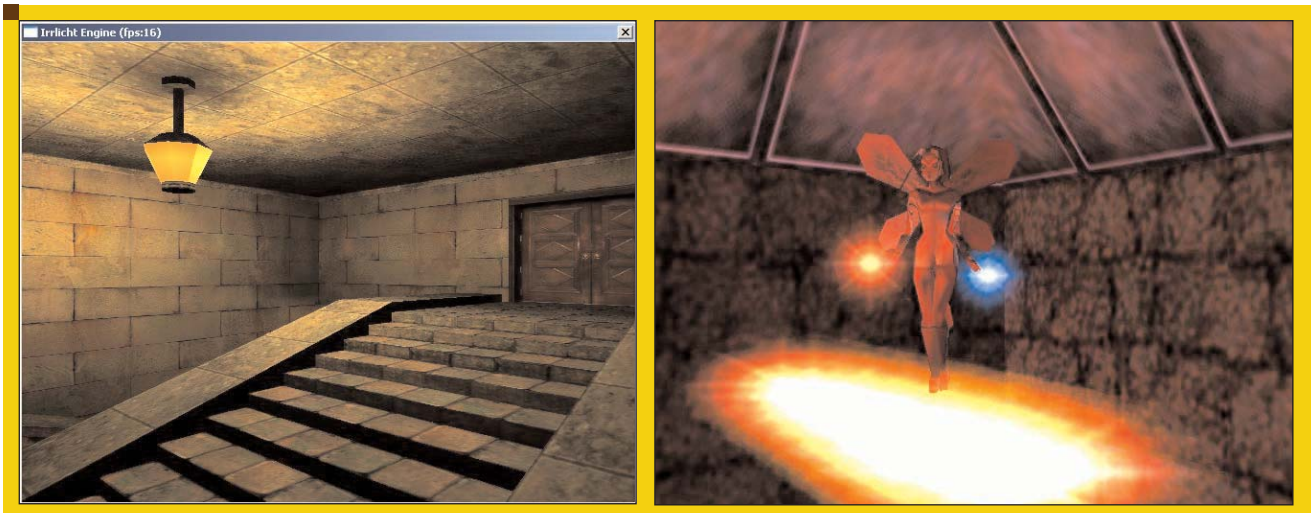


A körülötte látható téglalap a legkisebb „doboz” amibe bele rajzolni a harcost. Ha azt akarjuk megvizsgálni, hogy egy nyílvessző eltalálja-e a harcosunkat, akkor is találatot fog generálni az algoritmus ha nem a figurát, hanem a szürkére színezett területet találnák el. Ez sajnos nem jó. Ráadásul jelentős esélyünk van arra is, hogy egy ilyen „dobozolt” modell a pályán található ajtók 80%-án fenn fog akadni. A fenti példa a dobozolós algoritmus két dimenziós változata. Ehhez a primitív algoritmushoz képest jelentős előrelépés lehet valamilyen egyszerű sugárkövetési (***raytracing***) algoritmus használata. A sugárkövetési algoritmusok működése rettenetesen egyszerű, csak egy picit kell a geometriához érteni. Tételezzük fel, hogy van egy sugár (egyenes) amely keresztül halad „A” és „B” pontokon. Értelemszerűen a két pont távolsága az „A” és „B” pontokat összekötő szakasz hossza. Tegyük hát a következőt: a vizsgálni kívánt mozgó objektumunkból indítsunk egy egyenest. Az egyenes iránya az objektum haladási iránya. Vizsgáljuk meg, hogy milyen távolságra van a legközelebbi objektum amit el tudunk metszeni ezzel a sugárral. Ha találunk ilyen felületet akkor állapítsuk meg, hogy milyen távol van! Ha a távolság 0 körül van akkor az objektum nem haladhat tovább. Ha egy egyenessel akarunk elmetszeni sok kis háromszöget (poligont), akkor nincs más teendőnk mint egy nagy egyenletrendszerbe irni az egyenes irányvektoros egyenletét és a sík egyenletét, majd megoldani az egyenletrendszert. Az egyenlet gyökei a metszéspontot adják eredményül. Akit pontosan érdekel a dolog működése, látogasson el a ➲ ***http://www.lighthouse3d.com*** oldalra és az ***OpenGL***-es leírások között bőven találhat forráskóddal megtámogatott példát a fenti probléma megoldására.

A fenti algoritmus már igen jó hatásfokkal működik és jól optimalizálható. A ***Quake*** motor is ezt az algoritmust használja. Ha valaki tovább szeretné optimalizálni még ezt az algoritmust is, akkor javaslom a ***„térnyolcadoló fák” (Octree)*** utáni nézelődést az interneten. Az ***Octree*** egy nagyon hatékony eljárás, akár 70%-al is meggyorsíthatja a keresést. A lényege, hogy nyolc ágú fákat hoz létre úgy, hogy a modellteret nyolc részre osztja, majd a részeket ismét nyolc részre bontja, és így tovább. Ha felosztottuk a teret akkor már gyorsan meg tudjuk nézni, hogy az adott tér részben melyik objektumok vannak, az összes többit figyelmen kívül hagyhatjuk a keresés folyamán. Aki komolyan bele szeretné magát ásni a játékfejlesztésbe, annak javaslom a cikk végén található linkgyűjteményt, illetve a most következő ismertetőt az ***Irrlicht3D*** nevű játék motorról.

Irrlicht3D

Az ***Irrlicht*** egy teljesen ingyenes, nyílt forráskódú, stabil és platformfüggetlen ***3D*** motor azok számára akik nem



8. ábra Néhány kép – Ilyen az Irrlicht munka közben

érzik még magukat annyira bátornak, hogy egy saját motor fejlesztésébe fogjanak, de szeretnének belevágni a játékfejlesztésbe. Pár hónappal ezelőtt jutott eszembe, hogy megnézzem hol is tartanak most a nyílt játékmotor fejlesztések, ezért ellátogattam a ***Crystal Space*** oldalára. (A ***Crystal Space*** egy szintén nyílt motor, rengeteg extra szolgáltatással). Örvendezve láttam, hogy hamarosan elkészül a végső, stabil 1.0 verziószámú kiadás. Gondoltam belenézek, megnézem mennyit fejlődött mióta utoljára láttam. Már akkor picit zavarba jöttem amikor forráskód címen egy majd 40 Mbyte-os állomány kezdett el letöltődni hozzám, de amikor kibontottam és megláttam a forráskódját, majdnem leestem a székről.
Rengeteg szolgáltatással, kiegészítővel rendelkező motor, tényleg mindent tud ami szem-szájnak ingere, de programozó legyen a talpán aki játékot kezd fejleszteni vele.
Ne szaporítsuk tovább a szót: komplex és ehhez mérten nagyon bonyolult is. Ezt a kis incidenst követően ajánlották egy fórumon az Irrlichtet. A CS-es kalandomat követően kicsit bátortalanul fogtam neki a keresgélésnek. Az ➲ ***irrlicht.sourceforge.net*** oldalon rá is találtam a keresett enginere és meglepve tapasztaltam, hogy „csak” 14 Mbyte a forráskód ami tartalmazza a dokumentációt és 15 példaprogramot teljesen az alapoktól. Letöltöttem, kicsomagoltam majd ismét meglepődtem. Ennyire jól áttekinthető, jól strukturált és dokumentált programot keveset találni.

Az alábbi dolgokat tudja a 0.12.0-s verzió:

- Platformfüggetlen megjelenítés (***OpenGL*** vagy ***Direct3D***)
- ***Pixel*** és ***vertex shader*** támogatás
- Beltéri és kültéri „jelenetek” teljes támogatása, észrevétlen átmenet közöttük.
- Csontváz vagy „morph” alapú karakter animáció támogatása
- Dinamikus-fény kezelés, környezet tükröző textúrák, tűz, víz, időjárás effektek
- Részecske rendszer (robbanások)
- Komplett ***2D***-s grafikus felület, a saját programunk kezelőfelületének kialakításához.
- Jól dokumentált, sok példával ellátott egyszerű dokumentáció
- Elterjedt modell formátumok támogatása: ***Maya (obj)***, ***3D Studio Max (3ds)***, ***COLLADA (dae)***, ***DeleD (dmf)***, ***Milkshape (ms3d)***, ***Quake 3*** pályaszerkezet ***(bsp)***, ***Quake 2*** karakter modellek ***(md2)***
- Elterjedt kép formátumok támogatása: ***bmp, png, psd, jpeg, tga, pcx...***
- Gyors és egyszerű ütközés vizsgálat.
- Közvetlen olvasás ***zip*** állományokból.
- *XML* parser
- *GCC 3.2+* támogatás

Ebből a listából egyértelműen kiderül, hogy az ***Irrlicht*** inkább „csak” egy *3D* engine mintsem teljes játék motor. Hiányzik belőle a hang- és hálózat kezelés, valamint nincsenek benne fizikai eljárások és mesterséges intelligencia sem. Ennek ellenére mindenkinek ajánlom figyelmébe aki eddig nem mert, vagy nem tudott belevágni a játékfejlesztés – nem is annyira – „misztikus” világába.
Végezetül álljon itt néhány kép amelyek az Irrlicht példaprogramjaiból vannak, valamint a cikkben említett linkek, leírások.

Tóth Péter
(thotacc@drotnet.hu)
A BMF hallgatója vagyok, mellette egy kis- és középvállalatok informatikai rendszereinek Linuxos átállításával és szoftverfejlesztéssel foglalkozó cég informatikai vezetőjeként tevékenykedem. Kevés szabadidőmet barátnőmmel és barátaimmal töltöm egy-két sör társaságában.

KAPCSOLÓDÓ CÍMEK

Az Irrlicht3D honlapja
➲ http://irrlicht.sourceforge.net

OpenGL példa programok, leírások, érdekességek
➲ http://nehe.gamedev.net

OpenGL, VRML, Sugárkövetés
➲ http://www.lighthouse3d.com

Egy nyílt forráskódú fizikai engine
➲ http://ode.org/

Egy másik nyílt fizikai motor
➲ http://www.newtondynamics.com

XML Túra (1. rész)
Alapok

Egy újabb rövidítés, mely talán még sokak számára nem világos, hogy mire is hivatott. A cikksorozat első részében lerántjuk a leplet erről az eszközről, megnézzük honnan jött és miért, valamint az alkalmazásaira is fordítunk majd egy kis figyelmet. Lássuk hát miről is szól az Extensible Markup Language, azaz az XML.

Mit takar a név?

Magyarul nevezhetjük ***kiterjeszthető leíró nyelvnek***. Ez az eszköz a ***W3C (World Wide Web Consortium)*** keze munkája, egy általános célú jelölő nyelv, melynek segítségével sajátos, speciális célú leíró nyelvek konstruálhatók. Az ***SGML*** nyelvek közé sorolhatjuk. Mivel igen alkalmas különböző adattípusok leírására, ebből is adódik, hogy elsődleges célja strukturált adatok megosztása az interneten keresztül (***RSS, SVG***).

A történet

A fejlesztés alapjai egy ember nevéhez fűződnek: ***Tim Bray***. Mielőtt a ***W3C***-hoz került volna, egy az ***IBM***, ***Oxford University Press*** valamint a ***University of Waterloo*** által támogatott projekten dolgozott. A projekt egy internetes szótár kialakítását tűzte ki célul. Nagy hangsúlyt fektettek azon módszerekre, melyeket az adatok indexelésére és tárolására alkalmaztak. Ebben a struktúrában beágyazott címkék adták meg az adat fajtáját, így született meg az ***XML*** elődje. Később ***Tim Bray*** lett a ***W3C XML*** specifikációjának szerkesztője.

Miért is volt szükség az XML-re?

Az internet fejlődésével egyre több, bonyolultabb struktúrájú adatot kívántak közzétenni. A ***HTML***-ben viszont egyre nehézkesebbé vált az ilyen bonyolult struktúrák ábrázolása. Szükségesség vált egy szabadabb nyelv megalkotása. Míg a ***HTML*** csak egy megadott elemhalmazt használhat fel, addig az ***XML***-ben magunk alkothatunk elemeket. Nem tartalmaz annyi megkötést, mint az ***SGML*** nyelvek általában, ezért könnyen megtanulható, programozható és olvasható. Az ***XML*** így lett a ***HTML*** kisegítője, együtt jelentős mértékben növelték a weblapok lehetőségeit.

Célkitűzések

Az ***XML*** sajátos filozófiával rendelkezik. A fejlesztők több dolgot vettek figyelembe munkájuk során. Mindenek előtt a könnyű használatot, hogy gyorsan elterjedjen a nyelv, és ezzel egyetemben a széleskörű felhasználhatóságot is szorgalmazták. Az opcionális elemeket próbálták minimálisra csökkenteni, mivel ezekből eredhet a legtöbb kompatibilitási probléma. A forráskód kinézeténél ügyeltek a világos és olvasható struktúrára. A nyelvet minél előbb el kellett készíteni, nehogy esetleg több (kisebb cégek próbálkozásaiból származó) nyelv terjedjen el ezzel a céllal. Nagy jelentőséget tulajdonítottak az olvashatóságnak és ezért nem is törekedtek nagyon a tömörségre.
A tervük – úgy tűnik – bevált. 1996 óta az ***XML*** futótűzként terjedt el a webfejlesztők, programozók körében és ma már igen sok helyen felüti a fejét. Nem egy mobiltelefon használja bizonyos adatok tárolására. Faszerkezetű struktúrája alkalmas bármilyen (logikusan csoportosított) adat tárolására, mint például táblázat, címjegyzék, üzleti tranzakciók vagy műszaki rajzok. Az adatok stíluslapok (***CSS***) alkalmazásaival könnyen megjeleníthetővé válnak. Egyedi dokumentumtípus létrehozásával kialakítható akár saját jelölő nyelv is: ***WML*** (Wap oldalak leírására), ***VML*** illetve ***SVG*** (vektorgrafika tárolására) valamit ***OFX*** (pénzügyi információcserére). A sort még jócskán lehetne folytatni. Az ***OpenOffice*** és az ***AbiWord*** is natív fájlformátumként az ***XML***-t alkalmazza. Népszerűsége mellett szól az is, hogy támogatja a ***UNICODE***-ot és ezáltal lehetővé teszi bármely információ bármely emberi nyelven történő közlését. Képes a legtöbb számítástudományi adatstruktúra ábrázolására, mint például ***fa***, ***lista***, ***rekord***. Könnyű olvashatóságát ***öndokumentáló*** tulajdonsága is javítja, struktúra és mező neveket ír le a hozzájuk tartozó esetleges értékekkel együtt. Egyszerű szöveges formátumként valósul meg, így következik, hogy ***platform független*** is tehát nem nagyon reagál a technológiai változásokra sem. Ez a tulajdonsága is igen kecsegtető, hiszen ki is szeretné, hogy a ma elkészített dokumentumai holnap már olvashatatlanok legyenek valami kompatibilitási probléma folytán.

Valamit valamiért

Azonban, mint minden más dolog esetén, a kényelemnek és kezelhetőségnek itt is akadnak árnyoldalai.

1. Lista Egy gyökérelem

```
<dolgozók>
  <dolgozó>Kovács
Elek</dolgozó>
  <dolgozó>Szabó Elemér
</dolgozó>
</dolgozók>
```

2. Lista Két gyökérelem már nem megengedett

```
<dolgozók>
  <dolgozó>Kovács
  ➥Elek</dolgozó>
  <dolgozó>Szabó Elemér
  ➥</dolgozó>
</dolgozók>
<dolgozók>
  <dolgozó>Sándor
  ➥Tamás</dolgozó>
</dolgozók>
```

3. Lista Attribútum nevek alkalmazása

```
...
<dolgozó telefon="555-634"
➥kor="45">Kovács
➥Elek</dolgozó>
...
```

Az olvashatóság első nagy ára a tárolási költség megnövekedése. Főleg fontos lehet ***PDA***-k és mobilok esetén, mely eszközök előszeretettel alkalmazzák az ***XML***-t videók és képek leírására. Tömörítéssel ez a probléma még orvosolható, mint ahogyan például azt az ***OpenOffice*** is teszi. Nincs lehetőség a dokumentum egyes részeinek közvetlen elérésére vagy frissítésére sem. Feldolgozásában problémát jelenthet, hogy nincs valami gazdag típus kezelése, például nem oldható meg XML-ben hogy a ***Pi*** közelítő értékét lebegőpontos számként tároljuk karakterekből álló karakterlánc helyett.

Helyes XML dokumentum

Láttuk miről is szól az ***XML***, és remélhetőleg több olvasónak is viszket már a tenyere az alkalmazására.
Az ***XML nem*** egy ***programnyelv***. Használatához nem kell nagy programozási ismeretekkel rendelkezni. Könnyen és gyorsan megtanulható. Túránkat folytassuk azzal, hogy mikor beszélünk helyes ***XML*** dokumentumról. ***Helyes XML dokumentumról van szó, ha az helyesen formázott és érvényes.***

Helyesen formázott dokumentumok

Egy XML dokumentum csak egy gyökérelemet tartalmazhat. Például tekintsük egy fiktív munkahely dolgozóinak listáját feldolgozó dokumentumot (*1. Lista*).
Ez a példa egy gyökérelemet tartalmaz, amelynek neve `dolgozók`.
Ha most egy új `dolgozók` elemet szúrnánk be a *2. Lista* szerint, akkor a dokumentumunk már nem lenne helyes.
A nem üres elemek mind ***nyitó*** és ***záró*** címkével kell hogy rendelkezzenek.
Az *1. Listában* láthatjuk hogy a `dolgozó` elemhez tartozó nyitó tag `<dolgozó>` és a záró tag `</dolgozó>`.

Az üres elemek megjelölhetők ***önlezáró címkével***: `<üres/>`, ami persze megegyezik a következővel: `<üres></üres>`.
Minden attribútum értékének idézőjelek között kell szerepelnie, hogy szimpla (') vagy dupla (") az tetszőleges. Ami viszont lényeges, hogy szimpla csak szimplát, dupla idézőjel pedig csak duplát zárhat. Attribútumok alkalmazásánál érdemes beszédes neveket alkalmazni *3. Listát*.
A tag-ek egymásba ágyazhatók, de nem fedhetik át egymást. Tehát ha egy tag egy másik elemen belül kezdődik, akkor azon belül is kell lezáródnia.
A nevek kisbetű-nagybetű érzékenyek tehát figyeljünk oda, hogy például a: `<Példa>...</Példa>` helyes, amíg a `<Példa>...</példa>` helytelen használat.

Érvényes dokumentumok

Érvényes ***XML*** dokumentumok a felhasználó által definiált tartalmi szabályoknak megfelelő dokumentumok. Ilyen szabályrendszert alkothatunk akár ***DTD (Documentum Type Definition)*** segítségével is, melyet a dokumentumhoz csatolhatunk. Később még lesz szó a ***DTD***-ről és annak alkalmazásáról. A szabályok megszabják a helyes adatértékeket és azok helyeit. Például érvényességi szempontból hibás egy ***XML*** dokumentum, ha egy egész számként értelmezendő elem egy szöveges értéket tartalmaz, vagy éppen üres. Vannak olyan karakterek melyek nincsenek hatással az információ értelmezésére, ezeket nevezzük ***jelentés nélküli karaktereknek***. Ilyen karakter például ***tabulátor, szóköz, sortörés***.

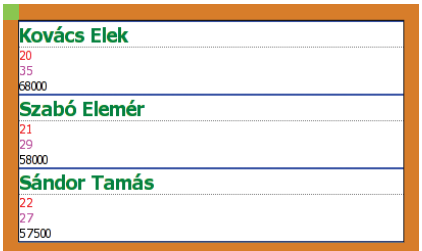
1. ábra A dolgozok.xml megjelenítése Firefoxban

Gyakorlat

Ennyit az elméletről. Itt az ideje, hogy alkossunk is valamit. Egy ***XML*** dokumentumot bármilyen szövegszerkesztővel létre tudunk hozni. A lényeg hogy szöveges állományként legyen elmentve „***XML***" kiterjesztéssel. Gyúrjuk tovább az először említett példát. Pár dolog feltűnhet, amiről még nem tettünk említést, de a *4. Listában* már szerepel. Az első sor tartalmazza a dokumentum ***fejlécét***. Itt adhatjuk meg, hogy mely ***XML*** szabvány szerint dolgozunk (***version***) valamint, hogy milyen karakterkódolást alkalmazunk (***encoding***). A második sor egy egyszerű megjegyzést tartalmaz. Csak hogy ez is egyértelmű legyen: minden megjegyzést `<!-` és `-->` jelek közé illesztünk. A frissen alkotott ***XML*** forrásunk máris megjeleníthető valamely böngésző segítségével. Nagyon újat nem fog mutatni a megjelenítés,

4. Lista Első XML dokumentumunk: dolgozok.xml

```
<?xml version="1.0" encoding=
➥"ISO-8859-2" ?>
<!- XML Túra 1. rész -
➥Dolgozók listája ->

  <dolgozók>
    <dolgozó>
      <név>Kovács Elek</név>
      <osztály>20</osztály>
      <kor>35</kor>
      <fizetés>68000</fizetés>
    </dolgozó>
    <dolgozó>
      <név>Szabó Elemér</név>
      <osztály>21</osztály>
      <kor>29</kor>
      <fizetés>58000</fizetés>
    </dolgozó>
    <dolgozó>
      <név>Sándor Tamás</név>
      <osztály>22</név>
      <kor>27</kor>
      <fizetés>57500</fizetés>
    </dolgozó>
  </dolgozók>
```



2. ábra CSS alkalmazásával Firefoxban

5. Lista Az adatok arculata CSS segítségével dolgozok.css

```
dolgozó
{
     display:    block;
     border:   1px solid blue;
     font-family: tahoma;
     width:  400px;
}

név
{
     display:    block;
     font-size:    16px;
     font-weight: bold;
     color:  green;
     border-bottom: 1px dotted
     ➥gray;
}

osztály, kor, fizetés
{
     display:    block;
     font-size:    10px;
}

osztály
{
     color:  red;
}

kor
{
     color:  magenta;
}
```

hiszen még nem definiáltunk semmilyen ***stílust***, mely szerint megformázhatná számunkra a böngésző az adatokat.

Stílus, forma, CSS

Amint már említésre került, az *XML* lehetőséget biztosít, arra hogy dokumentumunkhoz tetszés szerinti ***stíluslapot*** – akár többet is – hozzáfűzzünk. Ezek nélkül a böngésző csak egy ***fastruktúrát*** fog mutatni az állományból, melynek ágait általában be lehet zárni, vagy ki lehet fejteni tetszés szerint egy kattintással (*1. ábra*).
Az képkivágás tetején látható, amint a *Firefox* figyelmeztet is bennünket, hogy a megjelenített dokumentum nem tartalmaz stílusinformációt. Tüntessük hát el onnan azt a csúnya figyelmeztetést és vegyük át az uralmat dokumentumunk kinézete felett. Először létrehozunk egy *CSS* állományt mely tartalmazza a dokumentum és egyes eleminek a megjelenítési szabályait. A *CSS*-ről feltételezem nem először hall az olvasó. Ez a szabvány szintén a *W3C* ajánlása melyet főleg *HTML* dokumentumok formázására alkalmaznak. Egy sima szöveges állomány, melynek befűzését a dokumentumunk elején fogjuk megadni a böngésző számára.

Ha meg szeretnék szabni egy elem stílusát, akkor *CSS*-ben a nevével hivatkozunk rá, mint például ***osztály***. Aztán { és } jelek közé tesszük a formázás szabályait. Mivel a nem célom a *CSS* részletes magyarázata, ezért csak lényegretörően írok az egyes elemekről. A `display: block;` szabály adja meg az értelmezőnek, hogy az ezt követő elemek már másik sorba fognak kerülni. A `border: 1px solid blue;` adja meg hogy az elem körül egy 1 pixel vastag, folyamatos, kék vonal fog húzódni. A `font-size: 10px;` elég egyértelműen az elem betűtípusának méretét szabályozza 10 képpontra, míg a `color: red;` sor az elem betűjének színét pirosra állítja. Betűink vastagságát a `font-weight` tulajdonsággal tudjuk megadni, az elem szélessége pedig a `width` kulcsszó segítségével befolyásolható.
Már csak be kell kapcsolnunk a létrehozott stíluslapot a dokumentumunkba. Ez csak annyiból áll, hogy a fejléc alá beszúrunk egy hivatkozást:

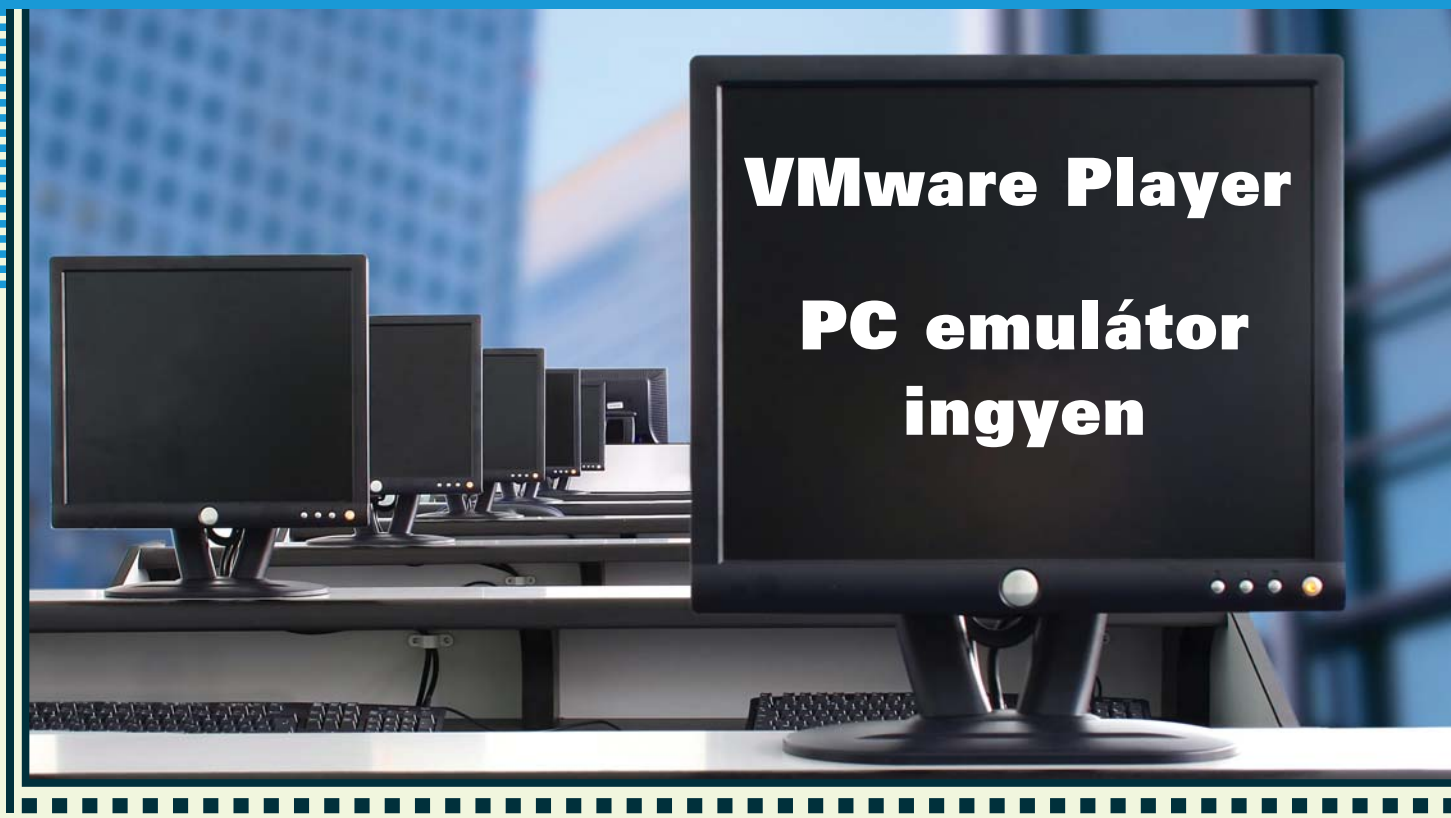
```
<?xml version="1.0"
➥encoding="ISO-8859-2" ?>
<?xml-stylesheet
➥type="text/css"
➥href="dolgozok.css" ?>
...
```

Innen fogja tudni a megjelenítő (böngésző), hogy az adott dokumentumon mely stíluslap szerint jelenítse meg.

Ami következik

Természetesen az *XML*-t nem arra találták ki, hogy szépen formázott adatokat jeleníthessünk meg a segítségével. Ennél komolyabb célokat tűztek ki a számára. A következő részben lesz szó az említett *DTD*-ről, *DSO (Data Source Objectről)* azaz arról, hogy hogyan is hivatkozhatunk például *HTML*-ből *XML*-ben tárolt adatokra.

Radics Péter
(peter.radics@gmail.com)
Az ELTE-n tanulok programtervező matematikus szakon. Hobbim a kosárlabda, autóvezetés, web-design, programozás. Főleg webes alkalmazások fejlesztése érdekel. 4 éve megrögzött Linux felhasználó vagyok.



VMware Player

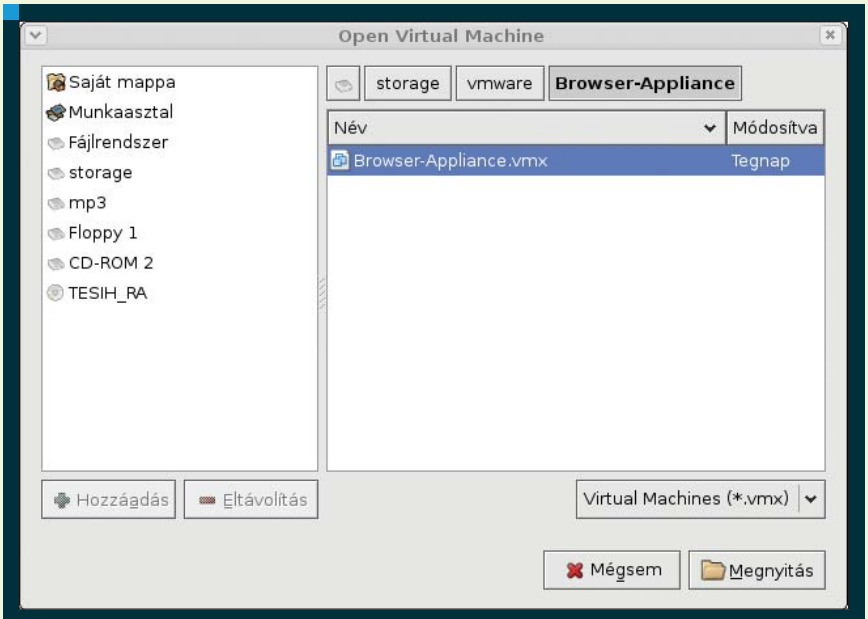
PC emulátor ingyen

A Linuxvilág magazin 2005 szeptemberi számában részletesen bemutattuk a VMware Workstation-t, mint az iparág egyik vezető virtualizációs szoftverét. Nemrég jelent meg a Workstation kistestvérének tekinthető Vmware Player, melyről jelen írásunkban adunk áttekintést.

Bár a *Workstation* szolgáltatás gazdag, stabil és keresztplatformos szoftver, ennek ellenére sok területről kiszorult, ilyen például az otthoni felhasználás. Mivel elsősorban vállalatok számára fejlesztett kereskedelmi termék, így otthoni felhasználóknak nehezen megfizethető. (Egy *Windows XP Home* árának közel kétszerese egy *Workstation* licenc)
De mire is jó egy ilyen szoftver az átlagembernek? Például amikor a felhasználó egy virtuális gépről böngészi a világhálót, teljes mértékben elkerülhetőek a különböző férgek és egyéb rosszindulatú algoritmusok támadásai mivel a *Vmware* teljesen elszigeteli a virtuális gépet a felhasználó valódi számítógépétől és operációs rendszerétől. Ezt és a virtualizációban rejlő számtalan hasonló lehetőséget felismerte a *Vmware* gyártója a *Vmware Inc*. is, ezért készített a *Vmware*-ből egy „butított" de ingyenesen hozzáférhető verziót a *Vmware Player*-t.

A *Player* mellet a gyártó honlapján közzétett egy virtuális gépet amin telepítve és megfelelően konfigurálva van az *Ubuntu Linux 5.10*-es változata és *Mozilla Firefox* (1.5 és 1.0.7 is).
Ezen szoftverek segítségével már a szűkebb pénztárcájú felhasználók is élvezhetik a virtualizáció előnyeit.
A *Player* jól jöhet számos más alkalommal is, például segítheti a vállalatok marketingtevékenységét.
Egy termékbemutató (feltéve hogy szoftverről van szó) végén az előadó mondhatja a hallgatóságának, akit részletesebben érdekel az adott termék az töltse le a *Player*-t és az előadó által elkészített virtuális gépet, amint előtelepítve található a termék, így nem kell telepítéssel konfigurálással vesződni a felhasználónak.
Ezen lehetőséggel már számos gyártó többek közt a *RedHat* a *Novell* az *IBM* és az *Oracle* élt. Termékeiket előtelepítették egy virtuális gépre, majd elérhetővé tették azokat az interneten.(Ezek egy helyen megtalálhatóak a *VMware Inc.* honlapján, és általában regisztráció után letölthetőek.)
A *Playert* elsődlegesen arra fejlesztették ki, hogy egy kész virtuális gépet tudjon használja a felhasználó, ezért nem tartalmazza az összes szolgáltatást amit a *Workstation*.
A legszembetűnőbb különbség a *Workstation* és a *Player* között hogy a Player nem képes virtuális gépek létrehozására, csupán a *Workstation*, a *Vmware GSX Server* és a *Vmware ESX Server* által létrehozott virtuális gépek elindítására, feltéve hogy azok a *Workstation 4*-es verziójánál nem régebbi szoftverrel lettek létrehozva. (A termék honlapja szerint a *Windowson* futó változata képes a *Microsoft virtuális gépek* és a *Symantec LiveState Recovery* lemez formátumú gépek elindítására is.)
Egyetlen megkötés, hogy csak egy processzort emuláló virtuális gépet tud elindítani a *Player*. (A *Workstation* legújabb verziójával (*5.5.0 build-18463*) már két processzort emuláló virtuális gépeket is lehet készíteni)
Ezen kívül a *Player* még más funkciókkal is rendelkezik, ezek a következők:

- A virtuális gépről elérhetőek a gazdagép CD/DVD meghajtói, hálózati kártyái és az USB eszközök



1. ábra Megnyitás ablak

- Megadható a virtuális gép rendelkezésére bocsátott memória mérete
- Hálózati képességek (Lásd a virtuális hálózatról szóló keretes írást)
- Virtuális gép leállítása szabályozható
- Jól használható, áttekinthető grafikus felület
- Teljes képernyős üzemmód

A ***Player*** nem túl erőforrás igényes program, a hivatalos dokumentáció szerint 500 MHz vagy annál gyorsabb processzor és 256 MB vagy attól nagyobb méretű memória szükséges a futtatásához. (a cikk írójának nem volt módja meggyőződni ezen állítások helyességéről) A program helyigénye már nem ilyen „barátságos” 150 MB maga a ***Player*** és ezenfelül körülbelül 1000-1500 MB egy virtuális gép. Gazdarendszer (az az operációs rendszer amire a ***Player***-t telepíti a felhasználó) egyaránt lehet 32 és 64 bites ***Microsoft Windows*** vagy ***Linux***.
Ezzel szemben a ***Player*** – csakúgy mint a ***Vmware termékcsalád*** többi tagja – elvileg minden i386-os géptípuson futni képes vendégrendszert (az az operációs rendszer ami a virtuális gépen fut) „le tud játszani” legyen az akár 32 vagy 64 bites. (64 bites vendégrendszert csak akkor indít el ha a gazdarendszergép 64 bites processzorral rendelkezik)

Telepítés, beállítás

A ***Player*** két formában elérhető, külön, és a ***Workstation 5.5***-ös változatával együtt is. A cikk írója az előbbit választja a telepítés lépéseinek bemutatásához. (Ezenkívül nincsen eltérés a két változat között)
A ***Player*** telepítésének előfeltétele hogy a felhasználó gépén telepítve legyen a ***gcc*** a ***make*** a ***perl*** és a futó rendszermaghoz tartozó fejlécállományok (header). (Ha az olvasó a ***Linux*** disztribúciójával szállított kernelt használja, akkor az ehhez tartozó headereket vagy külön csomagban, vagy a kernellel egyben találja meg.)
A ***Player*** ***tgz*** és ***rpm*** formátumban is elérhető. Én a továbbiakban a ***tgz***-ből való telepítést mutatom be. A ***Player*** telepítéséhez a ***Vmware-player-1.0.0-18587.tar.gz*** fájlt kell letölteni, majd kicsomagolni a

```
tar -vzxf Vmware-player-1.0.0
➥-18587.tar.gz
```

paranccsal. Miután kicsomagoltuk váltsunk át a ***vmware-player-distrib*** könyvtárba, majd adjuk ki a

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

parancsot. Ezzel elindul a telepítés. Elsőként néhány kérdést tesz föl a telepítő ami arra vonatkozik hogy mit hova telepítsen és hogy mi hol található (például init szkript stb.), ezekre válaszoljunk értelemszerűen, általában jó az alapbeállítás. Ezután a megfelelő helyre másolja a különböző fájlokat majd megkérdezi hogy kívánjuk-e most konfigurálni a ***Player***-t, válasszunk belátásunk szerint.
Ha nemet választottunk akkor a

```
vmware-config.pl
```

parancs futtatásával lehet elindítani a konfiguráló szkriptet.
Mivel az `rpm` nem végzi el a konfigurálást a továbbiakban a csomagból telepítőknek is mindent a cikkben leírtak szerint kell tenniük.
A konfigurálás során első dolgunk az lesz hogy elolvassuk és elfogadjuk a program felhasználói szerződését (licenc). Miután ez megtörtént ismét néhány fájl telepítési helyét kell megadni, általában megfelelő az alapértelmezett. Ezután rákérdez hogy lefordítsa-e a ***vmmon*** rendszermag modult, mondjunk igent. Ezt követően bekéri a ***C*** fordító (általában ***gcc***) és a kernel headerek elérési útvonalát. Ha ezeket csomagból telepítettük akkor a konfiguráló szkript valószínűleg helyesen határozza meg a helyüket, ha ez nem így van gépeljük be.
Ezután lefordul néhány kernelmodul, majd a hálózati szolgáltatások konfigurálására kerül sor.
Itt minden esetben megfelel az alapértelmezett nyugodtan nyomjunk mindig ***Entert***. A hálózati szolgáltatásokról a virtuális hálózatokról szóló keretes írás nyújt rövid áttekintést.
Miután beállítottuk a virtuális hálózatot is, megjelenik az utolsó kérdés, szeretné engedélyezni a ***Google*** kereső csíkot (***searchbar***) ? Ezt döntést az olvasóra bízom.
A Google kereső csík tulajdonképpen egy mező, ahova a keresendő szavakat kell begépelni majd Entert ütni. Ezután a futó virtuális gépen (ezt ***Windows*** és ***Linux*** alatt volt módom kipróbálni) automatikusan elindul egy böngésző és a ➲ ***http://www.google.com*** segítségével rákeres a megadott szavakra.
Ezzel kész is a telepítés és a konfigurálás, a `vmplayer` paranccsal lehet elindítani a ***Player***.
A ***Player*** a `vmware-uninstall.pl` parancs futtatásával lehet eltávolítani.

A virtuális hálózat

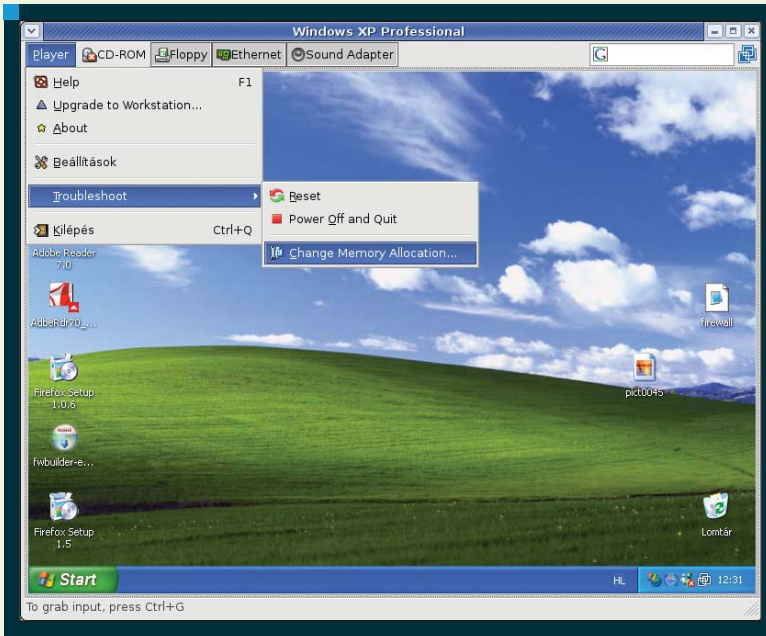
A *Vmware termékcsalád* igen kiterjedt hálózati képességekkel rendelkezik, képes emulálni egy teljes hálózatot de akár képes a virtuális gépet a fizikai hálózathoz kapcsolni. Virtuális gépeinket úgy csatlakoztathatjuk valamely hálózathoz hogy hálózati kártyát „szerelünk" bele. (Ezt akkor tehetjük meg amikor létrehozzuk a virtuális gépet. Mivel a *Player* nem ad erre lehetőséget, ezért itt nem lesz róla szó.)
A *Vmware Player* háromféle előre beépített hálózati topológiát tartalmaz. Ezek a következők:

- ***Bridged*** – Virtuális gépünk hálózati kártyáját egy híd (***bridge***) segítségével közvetlenül ahhoz a fizikai hálózathoz köti amelyikre gazdagépünk csatlakozik. Mivel virtuális gépünk fizikai hálózat részévé válik ezért a virtuális hálózati csatoló IP címét a fizikai hálózaton használt címtartományból kell kiválasztanunk. Virtuális gépünk külső fizikai gépekről is „látható" - azaz tudjuk pingelni. Mint minden hálózati kártyának, a virtuális csatolónknak is van fizikai címe (***MAC address***). Ezek a címek a gyártás során kerülnek kiosztásra, így elvileg minden fizikai cím egyedi. Mivel a *Player* 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 a gyakorlatban előfordulhat hogy címegyezés lép föl a hálózatba kötött két hálózati csatoló között, bár ennek kicsi az esélye. Ha ilyet tapasztal a felhasználó akkor a virtuális hálózati kártya MAC címét kell megváltoztatni, ehhez nyújt segítséget a ➲ ***http://www.vmware.com/support/ws55/doc/ws_net_advanced_mac_address.html*** weboldal.
- ***Host-only*** – Olyan hálózat amely egy gazdagépen belül létezik, teljesen elszigetelt a fizikai hálózattól, csak a gazdagépen futó virtuális gépek és a gazdagép csatlakozhat 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 gazdarendszerünkön az `ifconfig` parancsot a hálózati kártyák kilistázásához). Mivel ennek a hálózatnak nincs köze semmilyen fizikai hálózathoz ezért a virtuális hálózatunkon fut egy ***DHCP*** szerver (A *Player* gondoskodik róla) ettől kérjük az ***IP*** címet és egyéb hálózati információkat.
- ***NAT*** – A ***NAT*** és a ***Host-only*** hálózat közti különbség csupán annyi hogy a ***Host-only*** hálózatba kötött gépek képesek a külső hálózat elérésére is.

Utóbbi két lehetőséget csak a rend kedvéért mutattam be, a *Player* ugyanis egyszerre egy virtuális gépet képes elindítani ezért ezeket a képességeket csupán igen korlátozottan tudja használni.
A *Vmware* termékcsalád más tagjaival ezen a három gyári topológián kívül a felhasználó összeállíthat saját, teljesen egyedi hálózatot is, ezáltal lehetőség nyílik teljes hálózatok emulálására és tesztelésére.
A *Playerrel* csupán elég szegényes lehetőség nyílik a hálózati képességek kihasználására. Akit jobban érdekel ez a témakör az a ***Linuxvilág magazin 2005 szeptemberi*** számában megjelent ***Gép a gépen – Vmware*** című cikkben részletesebben olvashat róla.

Kezdjük el használni!

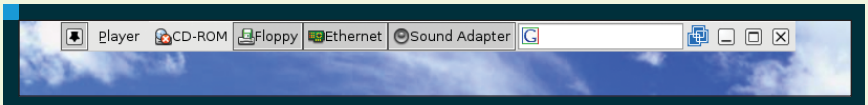
A *Player* indításához a `vmplayer` parancsot kell kiadni. Miután Entert ütöttünk megjelenik *Player* grafikus felülete és egy ablak amiben meg kell adnunk hogy melyik virtuális gépet szándékozunk elindítani. Ez utóbbi ablak teljesen hasonló egy szabványos Megnyitás ablakhoz. Miután megtaláltuk a keresett virtuális gépet a *.vmx* kiterjesztésű fájlt kell megnyitni. (*1. ábra*)
A megfelelő fájl kiválasztása után nyomjuk meg a megnyitás gombot ekkor elindul a virtuális gép.
A *Player* felhasználói felülete roppant egyszerű, a *2. ábra* mutatja. A *Player* menü *Preferences* menüpontjára kattintva megjelenik a *Preferences* ablak, itt az összes beállítás elvégezhető.
A legfontosabb a *Close Behavior* (viselkedés a kikapcsoláskor), két választási



2. ábra A Player felhasználói felülete

lehetőség van a ***Suspend the virtual machine*** és a ***Power off the virtual machine***. Előbbi azt jelenti hogy amikor a felhasználó kilép a ***Player***-ből akkor a virtuális gépet hibernálja, az utóbbi esetén pedig kikapcsolja. A ***Troubleshoot*** almenüben található a virtuális gép kikapcsoló gombja (ennek a viselkedését állítja a ***Close Behavior***) a ***Powr off and Quit*** és a virtuális gép ***Resert*** gombja. Ugyancsak a ***Troubleshoot*** almenüben van lehetőség a virtuális gép rendelkezésére bocsátott memória-mennyiség állítására. Ezt a ***Change Memory Allocation***-ra kattintva tehetjük meg. A memória méretét a megjelent ablakban a csúszka segítségével lehet állítani. Ahogy a vastagon szedett felirat is figyelmeztet ez a változás azután fog végbemenni miután újraindítottuk a virtuális gépet.

A ***Player*** menü mellet lévő gombok segítségével a gazdagépben lévő hardverelemeket (például floppylemez-meghajtó) tudjuk hozzákapcsolni illetve leválasztani a virtuális gépről. Ha például az előbb említett Floppyt hozzákapcsoljuk a virtuális géphez és egy lemezt behelyezünk a maghajtóba akkor az elérhető lesz a virtuális gépen futó operációs rendszerből is.
A címsorban lévő ***Maximie Window*** gomb segítségével lehet teljes képernyős üzemmódba váltani. Ekkor a teljes képernyőt a virtuális gépen futó operációs rendszer foglalja el. A képernyő tetején egy csík jelenik meg, erről a ***Player*** összes szolgáltatása elérhető. (lásd a *3. ábrát*) A ***Player*** menütől balra lévő gombbal lehet szabályozni hogy ez a csík állandóan látható legyen, vagy csak akkor jelenjen meg ha az egeret a felhasználó a képernyő fölső részéhez viszi.
Bár a ***Vmware Player*** nem túl bonyolult és nem sok mindenre képes, de mégis érdemes kipróbálni.



3. ábra Teljes képernyős üzemmód a Player grafikus felülete összezsugorodik

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

VMware inc honlapja:
➲ http://www.vmware.com/

Virtuális gépek letöltési helye
➲ http://www.vmware.com/vmtn/vm/

VMware Player honlapja:
➲ http://www.vmware.com/products/player/

VMware Player dokumentáció:
➲ http://www.vmware.com/support/pubs/player_pubs.html

Ismerkedés a LaTeX-hel (1. rész)

Jelenleg kevesen ismerik és használják Magyarországon a TeX rendszert. Pontosabban csak egy nagyon szűk kör, matematikusok, fizikusok, bölcsészek használják, viszont sok ilyen tárgyú folyóirat csak valamilyen, TeX alapú formátumban fogadja el a cikkeket. Tehát kevés ember használja, de ők igen intenzíven a TeX-et. Célom, hogy a TeX alapú rendszerek olyan képességeit mutassam be, melyek nem természettudósok számára is vonzóvá tehetik a TeX-et.

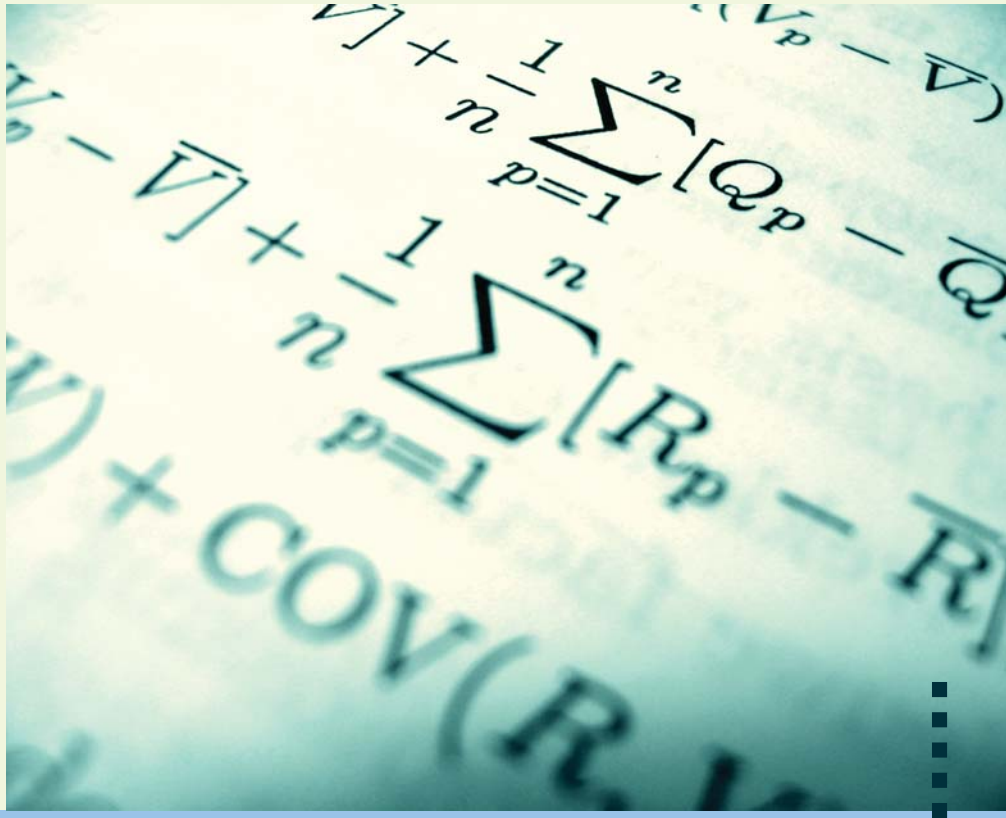
A *TeX* egy szövegszedő rendszer, amit *Donald E. Knuth* amerikai matematikus hozott létre az 1970-as években. Alapvetően angol nyelvű matematikai szövegek szedésére készült, ez *Knuth* több, nem túl szerencsés technikai megoldásán látszik. Ezeknek a megváltoztatása folyik, de már nem *TeX* néven, mert *Knuth* 1991-ben befagyasztotta a *TeX*-et. Azóta más néven folynak a fejlesztések, mint például *LaTeX*, *ConTeXt*, *Omega*. Részben ez a befagyasztás adja a *TeX*-re épülő rendszerek stabilitását.

Mivel a *TeX* nagy sikert aratott a természettudósok közt, bölcsészek is elkezdték használni, és saját igényeik szerint átformálni. Ezzel egyidőben elterjedt a nem latin betűs abc-vel író felhasználók közt is. Ez egyrészt a nyelvi képességek gyors fejlődésével járt, ugyanakkor nyilvánvalóvá váltak a Knuth által választott megoldások korlátai is. Részleges megoldást jelent a *Leslie Lamport* által írt makrócsomag, a *LaTeX*. Gyökeres változást a *Yannis Haralambous* által fejlesztett *Omega* rendszer fog hozni, ennek béta változatai már elérhetők. Az új rendszerben áttérnek az *Unicode*-ra, és jelentősen fejlesztik a *TeX* nyelvi képességeit.

Fontos tehát, hogy a *TeX* nem szövegszerkesztő, hanem egy olyan rendszer, ami karakter alapú szövegszerkesztőre építve nyújt tipográfiai és szövegszedői szolgáltatásokat. A *LaTeX* lehetővé teszi, és bizonyos fokig ki is kényszeríti a szerző és a tipográfus munkájának szétválasztását. Az elvek szerint a tipográfiai döntések meghozatala nem a szerző feladata, az a *LaTeX* dolga. A gyakorlatban ez az elv nem teljesül teljes mértékben, de jó közelítéssel igen. Nagyobb dokumentumok végső kinézetének megformálásában általában a szerző vagy a szerkesztő is aktívan részt vesz. Ezen elv egyik legjellemzőbb megjelenése a dokumentum hierarchiájának megvalósítása.

A szerző csak azt közli a *LaTeX*-el, hogy most egy új fejezet következik, az olyan teendők mint a betűtípus és méret megváltoztatása, helykihagyások, esetleg új oldal kezdése már a *LaTeX* feladata.

Ezen túl a *TeX* egy programnyelv is az összes megszokott lehetőséggel. Természetesen a szövegszedői képességek hangsúlyosak benne. Például egy *TeX* programmal kiszedhető az első száz prím garantáltan sajtóhiba nélkül. Ez azt jelenti, hogy a *TeX*



a	ا	a	d	د	d	.d	ض	z̤	m	م	m
b	ب	b	,d	ڈ	ḍ	.t	ط	t̤	n	ن	n
p	پ	p	_d	ذ	z̤	.z	ظ	ẓ	w	و	w
t	ت	t	r	ر	r	‘	ع	‘	,h	ہ	h
,t	ٹ	ṭ	,r	ڑ	ṛ	.g	غ	gh	y	ی	y
_t	ث	s̲	z	ز	z	f	ف	f	h	ھ	h
j	ج	j	ˆz	ژ	ts	q	ق	q	E	ے	ē
ˆc	چ	c	s	س	s	k	ک	k	’	ء	’
.h	ح	ḥ	ˆs	ش	ś	g	گ	g	T	ۃ	h
_h	خ	kh	.s	ص	ṣ	l	ل	l	.y	ۓ	ẏ
a	ـَ	a	i	ـِ	i	u	ـُ	u	.o	ـو	o̤
A	ـا	ā	I	ـِی	ī	U	ـُو	ū	.O	ـواَ	ō̤
.a	ـٔ	a̤	.u	ـٕ	u̇	o	ـۆ	o	e	ـێ	e
.A	ـأ	ā̤	.U	ـٔ	ū̇	O	ـو	ō	E	ـیے	ē

1. ábra Részlet a Klaus Lagally ArabTeX csomagjának kashmiri nyelvet bemutató dokumentációjából

program dönti el egy számról, hogy prím-e, és nem nekünk kell begépelni őket egy prímtáblázatból. A programozhatóság adta lehetőségeket hasznosabb célokra is lehet használni, az előbbi példa csak a *TeX* képességeit demonstrálta.

A *TeX*-el kapcsolatos információforrások közül a következő három érdemel föltétlen említést:

- *The Comprehensive TeX Archive Network*, ➲ *http://www.ctan.org/*
- *TeX Users Group*, ➲ *http://www.tug.org/*
- *Magyar TeX Egyesület*, ➲ *http://www.inf.unideb.hu/~matex/*

Ezek az oldalak jó kiindulási alapot nyújtanak a *TeX* megismeréséhez. Sok dokumentáció található rajtuk magyar, angol és más nyelveken is. A linuxos *teTeX* és a windowsos *MikTeX* is letölthető a ➲ *ctan.org*-ról, valamint kiegészítők, fontok, fejlesztői környezetek is elérhetők.

A TeX nem matematikai képességei

A *TeX* alapú rendszereknek számos olyan képessége van, melyek a nem matematikusok, természettudósok számára is érdekes, hasznos lehet. Ezek közül sorolok fel néhányat:

- ***Hosszú könyvek szerkesztése*** – A *TeX*-et tulajdonképpen erre találták ki. Több eszközzel támogatja a hosszú ideig tartó, csoportos munkát. A segítség egy része *TeX*-en belüli, további könnyebbséget jelent, hogy mivel a *TeX* txt alapú, használható az összes megszokott eszköz, mint például a *diff*, *cvs*, *rcs*. Ezek nagy részét a továbbiakban bemutatott *emacs* is támogatja.
- ***Nyelvi képességek – A TeX*** rendszert általában a matematikai képességei miatt ismerik, pedig a nyelviek miatt is érdemes használni. A már a *LaTeX* is sokrétű támogatást nyújt nagyon sok természetes nyelvhez, amit az *Omega* kiterjeszt. A fontokat biztosítja az összeshez, beleértve az arab, kínai, hindi és egyéb nem latin betűs nyelveket is. Valamint több mint 30 nyelven képes figyelembe venni az adott nyelvre jellemző tipográfiai szokásokat *(1. ábra)*.
- ***Játékok*** – Gó, sakk és bridzs játszmák szedésére is vannak csomagok, ezek közül egy gós példát mutatok *(2. ábra)*.
- ***Szimbólumok*** – A *ctan.org*-ról letölthető 105 oldalas ***symbols.pdf*** bemutatja a *LaTeX*-ben használható szimbólumokat, mint például ékezetek, pénznemek jelei, fonetikus abc-k, függvények, relációk, ezek tagadásai, biológiai, asztronómiai jelek.
- ***Bibliográfia*** – A *LaTeX* alapelveinek megfelelően itt is élesen szétválik a tartalmi és formai rész. Így könnyen alkalmazkodhatunk az egyes folyóiratok igényeihez, formai követelményeihez. Egyszer begépeljük vagy letöltjük a felhasznált cikkek bibliográfiai adatait, és utána ezt a bibtex a kívánt formátumban megjeleníti.
- ***Grafika – LaTeX*** dokumentumokba beilleszthetünk *PostScript* ábrákat, amiket elforgathatunk és át is méretezhetünk. További lehetőségeink is vannak ábrák készítésére. A legegyszerűbbeket elkészíthetjük *LaTeX* parancsokkal, a bonyolultabbakat például az *xfig* szerkesztővel. A grafikonok, mérési adatok, függvények kezelésére való ***gnuplot*** is képes együttműködni a *LaTeX*-hel.
- ***Automatikusan generált részek*** – A *LaTeX* egyik jellemzője, hogy hatékonyan támogatja az ábra és tartalomjegyzék, hivatkozások, indexek készítését. A forrásfájlban jelezzük, ha egy korábbi vagy későbbi részre hivatkozunk, és a többmenetes fordítás során a *LaTeX* beírja az oldal vagy fejezet számot.

Munka a TeX rendszerrel

A TeX-el végzett munka alapvetően három részre osztható. Ez az ismertető is ezt a felosztást fogja követni.

1. Egy szövegszerkesztővel elkészítjük a forrásfájl vagy fájlokat
2. A *TeX* programmal lefordítjuk
3. Egy megjelenítővel megnézzük az eredményt

majd módosítjuk a forrást, újra fordítjuk és így tovább. A munka elején megelégszünk azzal, hogy megközelítőleg elérjük a végső kinézetet, és inkább a tartalomra koncentrálva írunk. Majd a munka vége felé, mikor a tartalmi résszel készen vagyunk, következik a hibák kijavítása és a dokumentum végleges formájának megalkotása.

A forrásfájl létrehozása

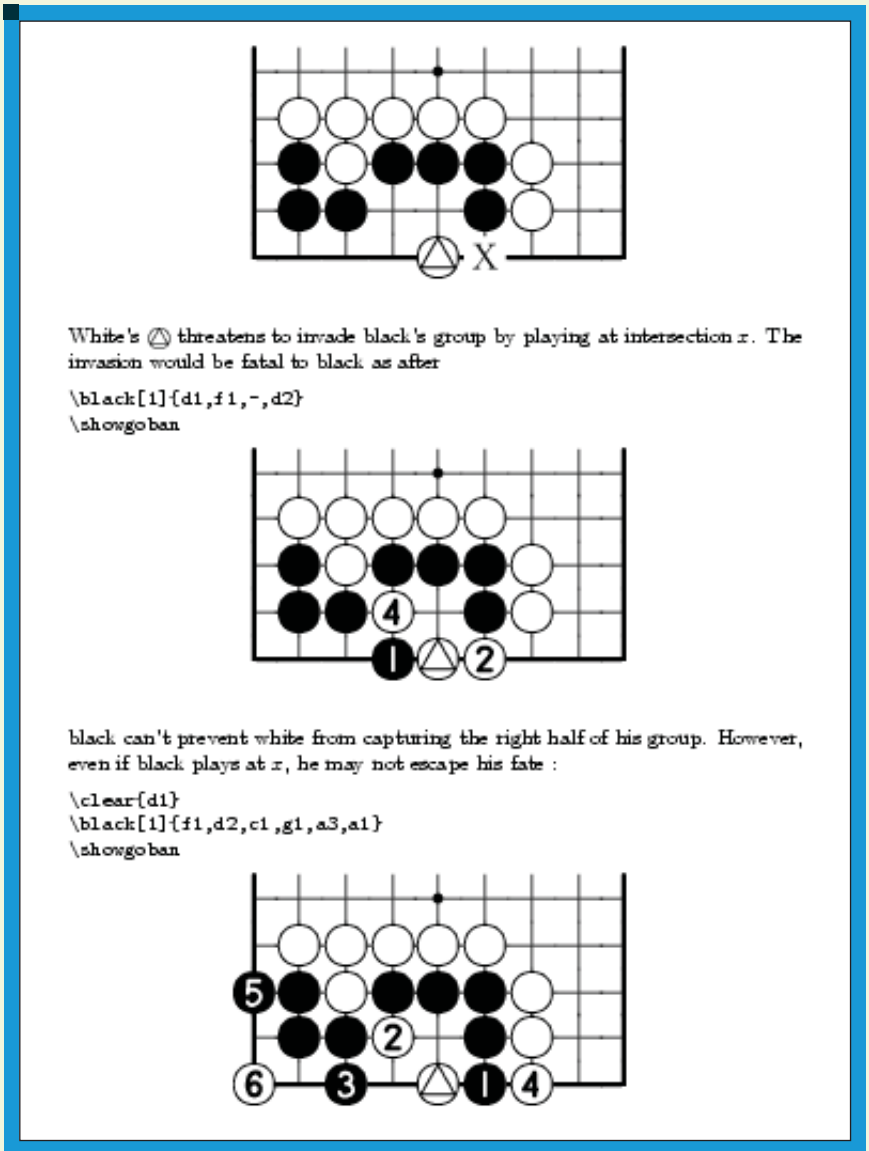
A forrás alatt azt a szövegfájlt értem, ami a bemenete a *LaTeX* és néhány egyéb programnak. Létrehozására bármely szövegszerkesztő megfelel ami képes ***egyszerű szöveg (plain text)*** formátumban menteni. Ennek ellenére érdemes olyant használni, amit úgy terveztek, hogy segítsék a *TeX*-el végzett munkát. Az egyik ilyen az *emacs*, abban fogom bemutatni a *TeX*-elés folyamatát.

Minden karakteralapú forrásfájl létrehozásában sokat segít, ha a szövegszerkesztő képes az éppen használt nyelv szintaktikáját, kulcsszavait felismerni, és annak megfelelően kiszínezni a forrást. Egy szövegszerkesztő még számos további szolgáltatást nyújthat, ezeket egy későbbi fejezetben írom le.

A ***fejlécben (header)*** kell deklarálnunk, hogy milyen csomagokat, saját parancsokat kívánunk használni az éppen szerkesztett műben. Általánosan elterjedt szokás, hogy mindenki készít magának egy fejlécet, amit állandóan használ, és nem írja át minden egyes *LaTeX*-ben írt műhöz. A jelenlegi számítógépek teljesítményénél a feleslegesen ott maradó utasítások, csomagbetöltések végrehajtása nem lassítja észrevehetően a fordítást, viszont nehézkes és lassú végiggondolni, hogy pontosan melyik csomag szedhető ki. Nagy könyvek esetén már érdemes ezt megtenni, de ott is inkább a csomagok közti inkompatibilitások miatt.

A fordítás menete

A *TeX* a fordítás során pontrendszert használ a legszebb eredmény eléréséhez. Például, egy bekezdés kiszedésekor több módon is megpróbálja elrendezni a bekezdést, és mindegyiket lepontozza, majd végül azt választja, amelyiknek legkedvezőbb a pontszáma. Egy pontozási szempont, hogy ne legyen túl sok elválasztás a sorok végén. Viszont, ha azt elérjük, hogy egyáltalán ne legyen elválasztás, akkor a szavak és betűk köze lesz nagyon egyenletlen, ami szintén nem szép. A pontozás a kettő közti kompromisszum megtalálásában segít. Természetesen a pontozás alapértelmezett értékeit saját ízlésünk szerint átállíthatjuk, így befolyásolva a dokumentum megjelenését, ez a mindennapi gyakorlatban nem szokás, nagyon jók az alapértelmezések.

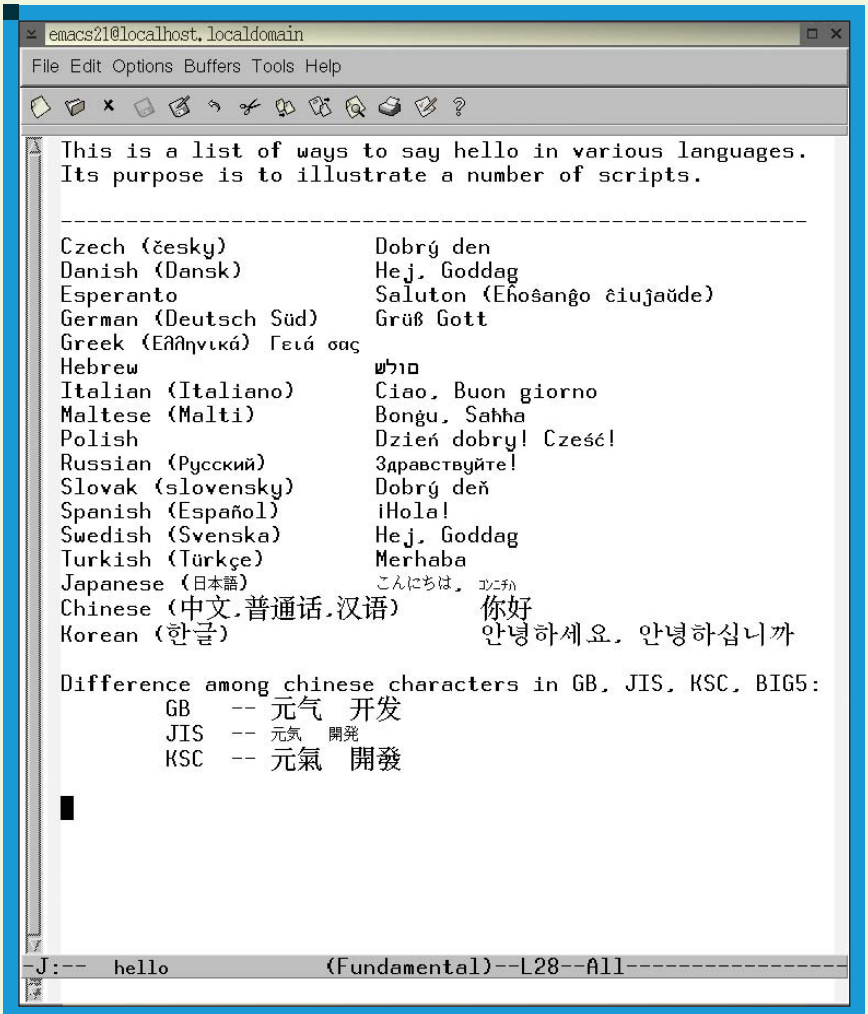


2. ábra Részlet az Étienne Dupuis által írt igo csomag dokumentációjából

A fordítás során létrejövő fájlok

Alapvetően csak néhány fájlra van szükségünk, arra ami a szerkesztett dokumentumunkat tartalmazza, és esetleg a bibliográfiára. Érdemes azonban foglalkozni a létrejövő további fájlokkal is. Ez a hibakeresésben is, és a *TeX* rendszer megismerésében is segít.

Egy dokumentum tartalmazhat hivatkozásokat és egyéb automatikusan generált részeket. Mivel az első fordítás során az erre vonatkozó információk még nem állnak rendelkezésre, ezért szükséges a többszöri fordítás. Az első során létrejönnek ezek a fájlok, a továbbiak során meg feldolgozásra kerülnek. Például a tartalomjegyzék generálása úgy történik, hogy az első fordítás során kiírja egy *toc* kiterjesztésű fájlba a fejezetcímeket, és azt, hogy hányadik oldalon találkozott vele. A második során már felhasználja ezt, de egy hosszú dokumentum



3. ábra Az emacs-szel kapott hello fájl Debian Linuxon

esetén a tartalomjegyzék beszúrása miatt elcsúszhatnak az oldalak, ezt küszöböli ki a harmadik fordítás. Ennél többszöri fordításra csak nagyon különleges esetekben lehet szükség. Ehhez hasonlóan működik az ábrajegyzék, táblázatjegyzék és az irodalmi hivatkozások kezelése is. Minden fordítás során létrejön egy ***naplófájl (log)***, ez a fordítóprogram üzeneteit tartalmazza, hibakeresésben hasznos.

Az Emacs és a TeX

Az *emacs* karakter alapú szövegszerkesztő. Kutatók, egyetemi oktatók, hallgatók számára készült. Viszonylag sok parancsot, billentyűkombinációt meg kell tanulni a kényelmes használatához, ennek fejében nagyon homogén rendszert kapunk. Felkínálja a tudományos életben szükséges formátumok, számítógépes nyelvek egyszerű és korrekt eredményt adó használatát, például cikkek, futtatható programok és e-mail-ek írását.
A program teljesen ingyenes: forráskóddal és dokumentációval együtt terjesztik *GPL* licenc alatt. Emiatt szinte minden más operációs rendszerre is átvitték, és lényegében garantáltnak tekinthető, hogy a jövőben is ez marad a helyzet. A *UNIX* világban nagyon elterjedt: az *emacs* és a *vi* a *UNIX* világ két legalapvetőbb karakter alapú szövegszerkesztője. A *LaTeX*-hez hasonlóan az *emacs* is támogatja a nem latin betűs írásokat, ezt a *3. ábra* következő kép mutatja.
Az *emacs*-ben a munka általában két, esetenként több lépésben történik. Az első szakaszban begépeljük a kívánt szöveget. A másodikban pedig fordítás, például *TeX* lefutása, után nézhetjük meg az elkészült munkát. A további lépések a hibakereséssel és javítással, mint például nyomkövetés, kapcsolatosak.
Minden kezdeti furcsasága ellenére érdemes megismerni, mert nagyon hatékony eszköz minden karakterorientált alkalmazáshoz.
A legtöbbhöz már írtak egy külön csomagot. Hatékonyságát a saját, szövegkezelésre optimalizált *LISP* nyelve biztosítja. A legtöbb *emacs* parancsot is az *emacs LISP* nyelvén írták meg.

Auctex

Az *auctex* csomag nagyon kibővíti az *emacs TeX*-el kapcsolatos képességeit. Lehetővé teszi, hogy a fejlesztés teljes menete az *emacs*-en belül történjen. Elég egy szövegszerkesztőt megszokni a szöveg beírásához, és a hibajavításhoz, sőt a már megszokott parancsokat használhatjuk a fájlkezelésben is.
Az *auctex*-et használva lehetőségünk van egerészni, ami a *TeX*-el végzett munka kezdetén minden bizonnyal kényelmes dolog. Ennek során mindig kiírja az *emacs* a kiválasztott menüponthoz tartozó billentyűkombinációt, amit előbb-utóbb megtanulva lényegesen gyorsul a munkavégzés.

Összefoglalás

A *LaTeX* első ránézésre bonyolult, összetett rendszer. Mégis érdemes időt és fáradtságot szánni a megismerésére, mert hatékonyan képes segíteni a nehéz szövegformázási feladatok megoldását. Köztük a bibliográfia és az automatikusan generált részek kezelését, valamint különösen az *Omega* használatával egy dokumentumon belül is kényelmesen tudunk nyelvet váltani. A *TeX* alapú rendszerek használatát jelentősen megkönnyíti egy jól beállított fejlesztői környezet. Mint például az *auctex*-hel kibővített *emacs*, ez támogatja a *TeX* használatának teljes folyamatát.

Vecsei Balázs
(vecseib@math.bme.hu)
A BME TTK-n végzett matematikusként. 1998 óta foglalkozik Linuxszal és LaTeX-hel. Szabadidejében sokat túrázik Magyarországon és külföldön is, kedvence a Börzsöny.

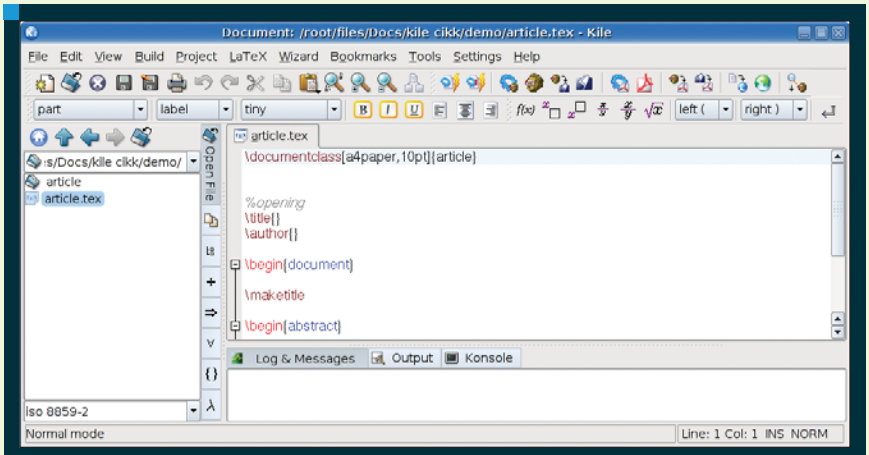
LaTeX szerkesztés könnyebben: Kile

A Kile egy LaTeX dokumentum-szerkesztést segítő olyan egységesített környezet, amely egy grafikus felhasználói felületbe zárva számos eszközt tartalmaz, megkönnyítve LaTeX dokumentumok létrehozását és szerkesztését.
Egy könnyen használható felhasználóbarát felület biztosításán túl beépített eszközöket tartalmaz LaTeX dokumentumok lefordítására – vagyis a forrás alapján történő végső dokumentum előállítására –, hibakeresésére, konverziójára és előnézetére.

Ebben az írásban próbát teszünk arra, hogy a számos funkció és lehetőség közül a legfontosabbakat bemutassuk, remélve, hogy azok az olvasók is, akiket eddig visszarettentett a *LaTeX* sokrétűsége és látszólagos bonyolultsága, kedvet kapnak és belevágnak a *LaTeX* dokumentum-szerkesztés világába.
A *LaTeX* környezet és nyelv bemutatásával egy másik, *LaTeX*-el foglalkozó sorozatban találkozhatnak.
A *Kile* egy nyílt forrású, alapvetően *Linux* alapú operációs rendszereken *KDE* grafikus felület alatt futó alkalmazás. Természetesen *Gnome* és egyéb grafikus felületek alatt is használható, ha a szükséges *KDE* könyvtárakat telepítjük. *Windows* operációs rendszerek esetén *Cygwin* alatt futó *KDE* felületen is futtatható. Ebben a bemutatóban a *Kile 1.8*-as verzióját használjuk, *Debian Sid GNU/Linux* operációs rendszeren, *KDE 3.4.2* grafikus felület alatt.

Ismerkedés, első lépések

A *Kile* szerkesztő használatához telepítenünk kell magát a *Kile* alkalmazást és a szükséges segéd-könyvtárakat, illetve egy működő *LaTeX* környezetet. A telepítés módja gyakorlatilag minden Linux disztribúcióban máshogy zajlik, *Debian GNU/Linux* és az erre épülő más disztribúciók alatt (Ubuntu, Mepis, Knoppix, Kanotix, stb.) a



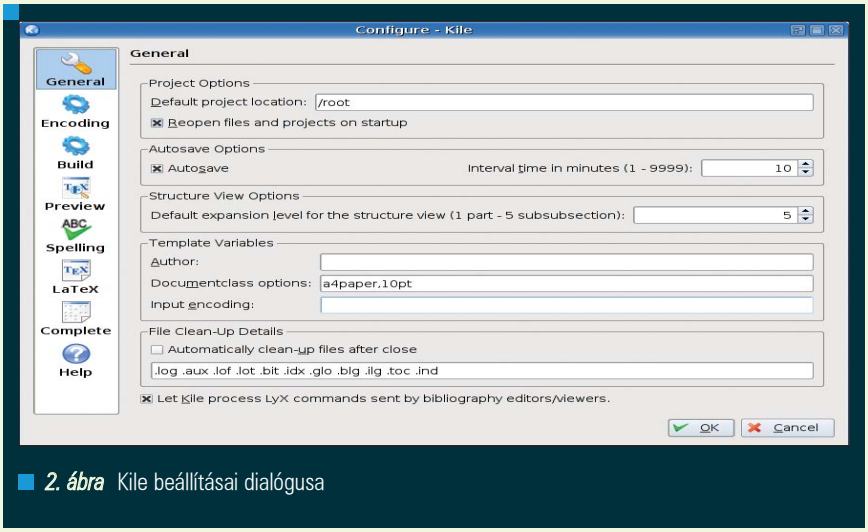
1. ábra A Kile főablaka egy üres, „article" stílusú dokumentummal


```
sudo apt-get install kile
```

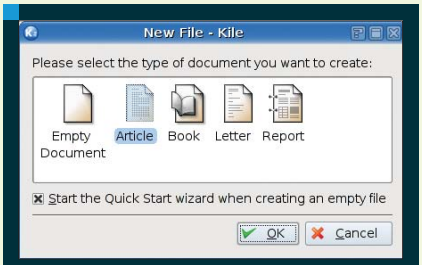
paranccsal a *Kile*-t és a szükséges segéd-alkalmazásokat és könyvtárakat is telepíthetjük. Egy működő *LaTeX* környezethez telepítenünk kell a *tex-common* csomagot és gyakorlatilag az összes *tetex-* kezdetű csomagot (például *tetex-base*, *tetex-bin*, *tetex.-extra*).
A *Kile* indításakor a főablak fogad minket, amely az *1. ábrán* látható.
A felhasználói felület három fő részből áll: az oldal-panel (az *1. ábra* bal oldalán), az üzenet-panel (az *1. ábra* alsó részén) és a forrás-szerkesztő.
Az első használat előtt érdemes a *Kile* beállításait végignéznünk, ami a *Settings* menü *Configure Kile* menüpontjából érhető el.

Ahogy a *2. ábra* is mutatja, számos tulajdonságot állíthatunk be, amelyek közül néhány nem létszükséglet, de néhány nagyon fontos lehet a későbbiekben, például a fájlok kódolásának beállítása (magyar szöveg esetén javasolt az *UTF-8* vagy *ISO 8859-2* kódolás használata), a forrásfájlok fordítási paraméterei, helyesírás-ellenőrzés beállításai, stb.
Új dokumentumot legegyszerűbben a *File* menü *New* menüpontjával hozhatunk létre, amely egyúttal lehetőséget ad a létrehozandó dokumentum alapstílusának meghatározására is (*3. ábra*).
Egy új dokumentum létrehozását a *Kile* több módon is lehetővé teszi.
A *Kile* támogat projekt keretében történő rendszerezést, új projektet pedig

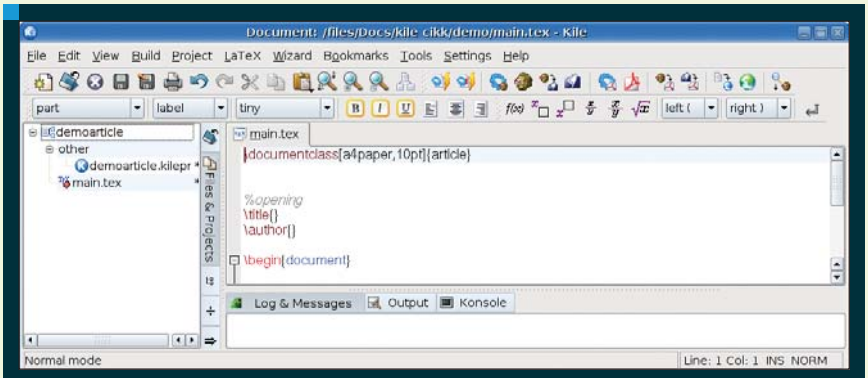
a *Project* menü *New Project* menüpontjával hozhatunk létre. A projekthez hozzáadhatunk már létező fájlokat, illetve a *File* menü *New* menüpontjával hozhatunk létre a projekthez tartozó új fájlokat. A projekthez tartozó fájlok listáját az oldal-panel segítségével gyorsan áttekinthetjük, ha a panelen a *Files&Projects* nézetbe váltjuk, ahogy az a *4. ábrán* látható. Amikor létrehozunk egy új dokumentumot és meghatározzuk a stílusát, akkor a *Kile* egy üres minta-dokumentumot hoz létre, amely tartalmazza az alapvető elemeket: alapstílus megadása, minimális dokumentumstruktúra (*5. ábra*).



2. ábra Kile beállításai dialógusa



3. ábra Új .tex fájl létrehozásának dialógusa



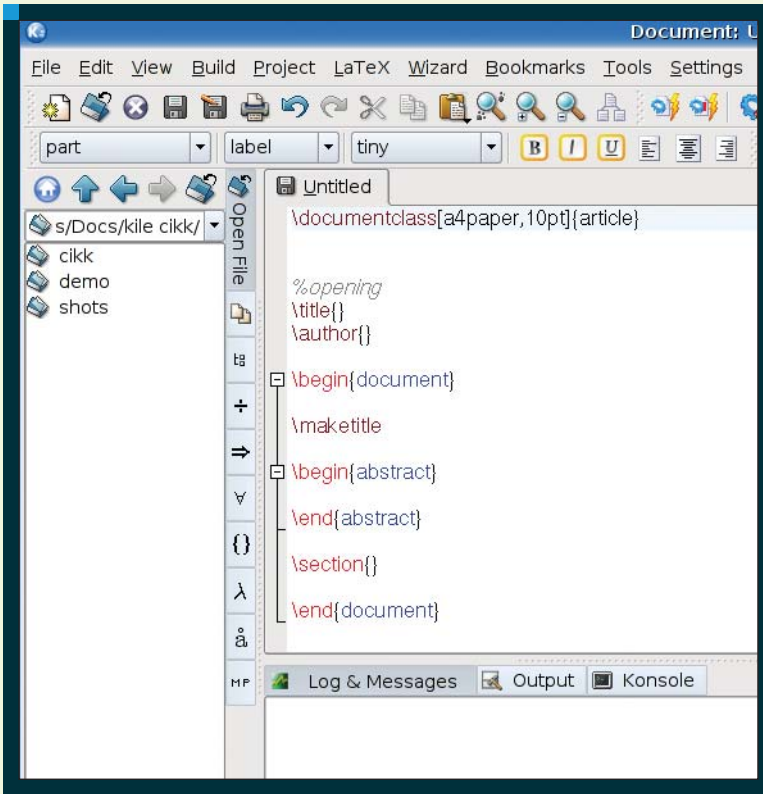
4. ábra Új projekt, egy main.tex nevű fájlal, projekt-nézettel az oldal-panelen

Az első dokumentumunk sikeres létrehozása után, mielőtt még egy dokumentum megírásához kezdenénk, először a szerkesztő tulajdonságaival ismerkedjünk meg.

A szerkesztő

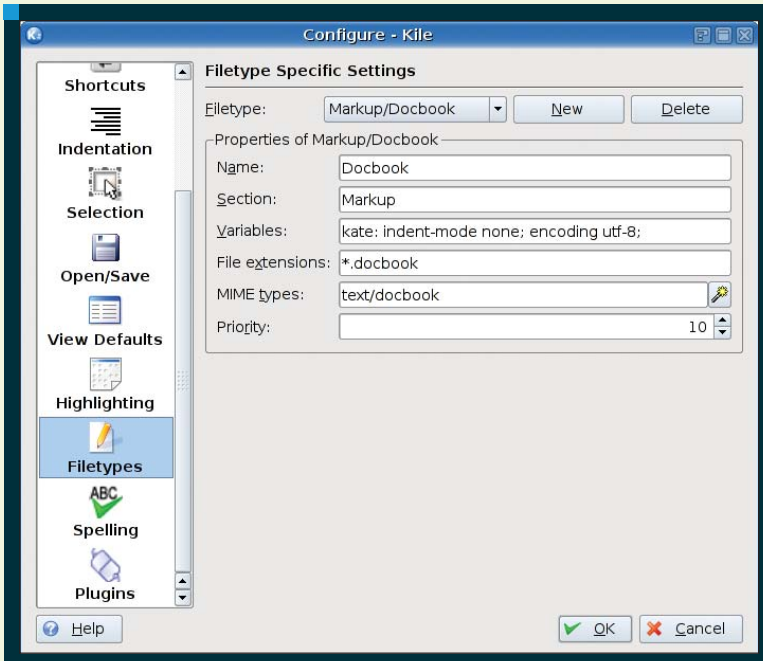
A Kile szerkesztője az a hely, ahol a dokumentumok készülnek, ezért érdemes elidőzni a tulajdonságai és képességei fölött, amelyek megkönnyíthetik a munkát. A szerkesztő tulajdonságainak beállítására is lehetőségünk van a *Settings* menü *Configure Editor* menüpontjában, ahogy az a *6. ábrán* is látható.

A szerkesztő számos olyan tulajdonsággal bír, amely nagyban megkönnyíti a *LaTeX* dokumentumok írását. Az egyik ezek közül a szintaxis kiemelése/színezése, amely könnyen áttekinthetővé teszi a forrást. Egy másik fontos tulajdonság, hogy egy `\begin{...}` elem beírása után automatikusan beszúrja az `\end{...}` elemet, ezzel nagymértékben csökkenhet a nyitva hagyott elemek

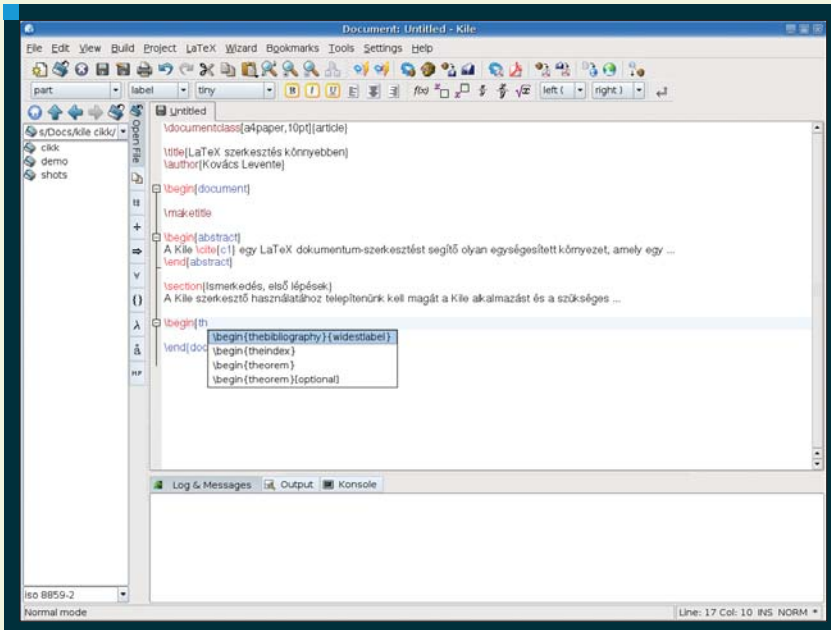


5. ábra Egy „article” típusú dokumentum létrehozásának eredménye

száma, ami leginkább sokszoros egymásba-ágyazottságnál fontos segítségnek bizonyulhat. Ezek mellett szintén fontos tulajdonság, hogy ha a forrásban mozgunk és egy nyitó vagy záró zárójelre (bármilyen típusúra) helyezzük a kurzort, automatikusan kijelöli a zárójel párját. *LaTeX* forrás-szerkesztés szempontjából egy létfontosságú, a munkát gyorsító tulajdonság az, hogy parancsok beírásakor már az első karaktertől felkínálja az addig beírt karakterekhez leghasonlóbb parancsok beszúrását. Esetenként csupán néhány karakter beírása elég a parancs beszúrásához, ez nagyon gyors szerkesztést tesz lehetővé. Az előbbi néhány tulajdonság látható a *7. ábrán* is.
A szerkesztő lehetőséget ad arra is, hogy a `begin-end` elemek által határolt forrás-részeket összevonhassuk és kibonthassuk, akár egymásba ágyazott struktúrák esetén is, ami hosszú forrásfájlok szerkesztésekor megkönnyíti a dokumentum áttekintését, ahogy az a *8. ábrán* is látható.
A szerkesztő működésének sokoldalúságát hatalmas mértékben növeli az oldalsáv (***Side Bar***, ***Settings*** menüben ***Show Side Bar*** menüponttal jeleníthető meg) illetve a számos, szerkesztést segítő funkció amelyek az eszköztárakból és a ***Kile*** menüiből érhetők el. A következőkben mindkettőre részletesen kitérünk.



6. ábra A Kile szerkesztő tulajdonságainak beállítási dialógusa



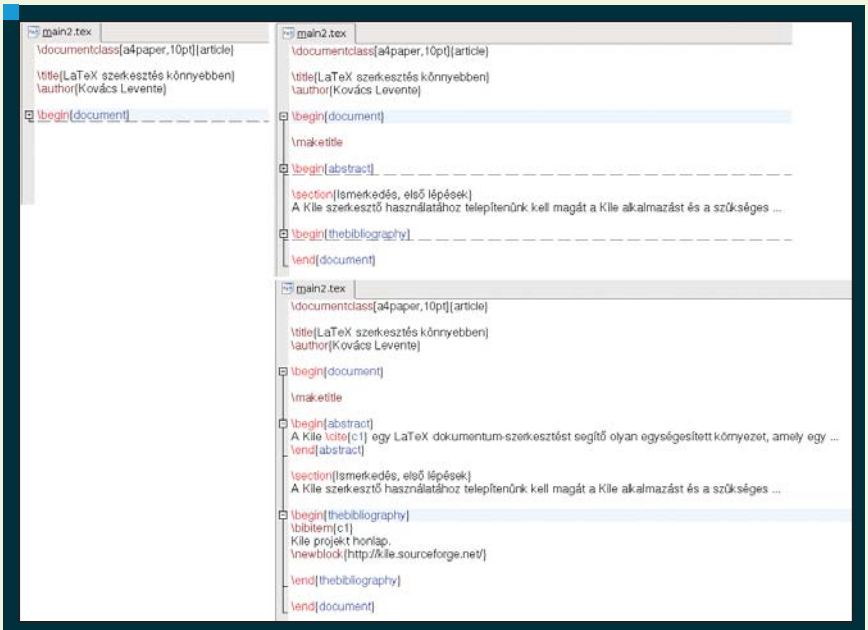
7. ábra A Kile szerkesztője: szintaxis-színezés, helyi Intellisense jellegű segítség a parancsok beírásánál

Az oldalsáv

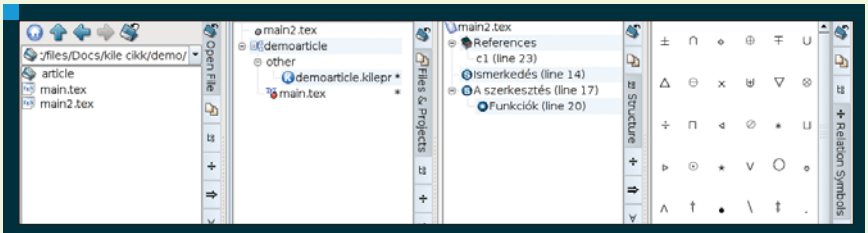
A ***Kile*** szerkesztést segítő eszközei közül az oldalsáv annyi funkcióval bír, hogy megéri külön fejezetet szentelni számára. Igazság szerint nem azért tesz jó szolgálatot, mert forradalmi újításokat, tartalmaz, mert ez nem így van, hanem azért, mert használható formában és egy helyen foglal össze olyan funkciókat, amelyek segíthetik a munkát.
A panel működése és kezelése ugyanúgy történik, mint a ***KDE*** grafikus felület alatt a ***Konqueror*** fájlrendszerböngészőben és egyes erre épülő alkalmazásban. Az egyes funkciók külön panelekre bontva érhetők el, kategóriákra bontva. Az egyik panel az ***Fájl megnyitása*** panel (***Open File***), amelyben a fájlrendszerünkön böngészhetünk és megnyithatunk fájlokat anélkül, hogy más alkalmazást kellene használnunk, vagy a ***File*** menü ***Open*** dialógusát kellene használnunk.
A ***Fájlok és projektek*** (***Files and Projects***) panelen a ***Kile***-ban megnyitott fájlok, projektek és az ezekhez tartozó fájlokat láthatjuk összegyűjtve, könnyen elérhető formában. A ***Struktúra*** (***Structure***) panelen az aktív megnyitott dokumentum szerkezetét láthatjuk fa struktúrában, fejezetek szerint lebontva, amelyekre kattintva az adott fejezetre ugorhatunk a dokumentumban. A következő panelek különféle szimbólumok gyors elérésére szolgánk (például görög karakterek, matematikai jelek, zárójelek, stb.). A *9. ábrán* látható a panelek egy-egy részlete.

Fontosabb funkciók és eszközök

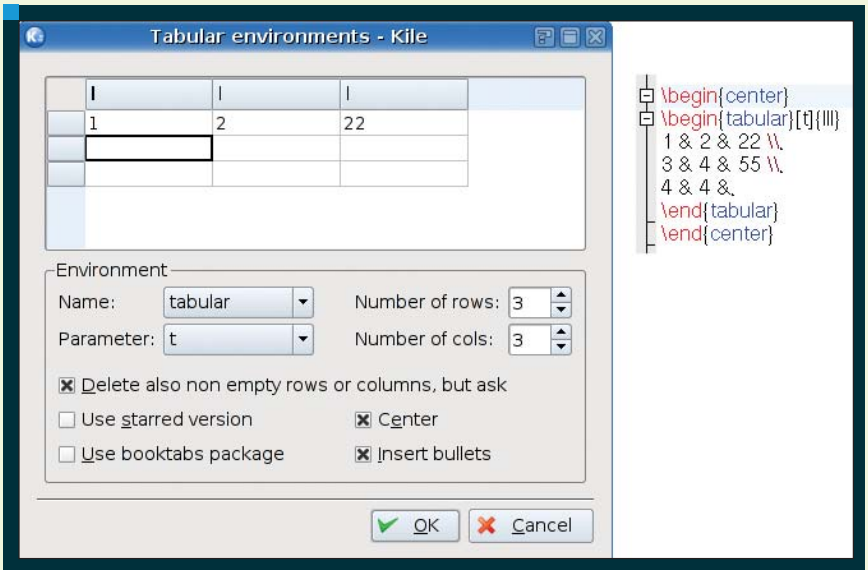
Miután a ***Kile*** felhasználói felületének fontosabb részeit megismertük, itt az ideje kicsit konkrétabban szerkesztési eszközökre kitérni. A ***Kile*** erőssége nem csak az eddig említett szerkesztést



8. ábra Struktúrák összevonása és kibontása a szerkesztőben. Bal felső: összevonva, jobb felső: a külső struktúra kibontva, jobb alsó: az összes struktúra kibontva.



9. ábra A Kile paneljei



10. ábra Táblázat definiálása és a generált kód

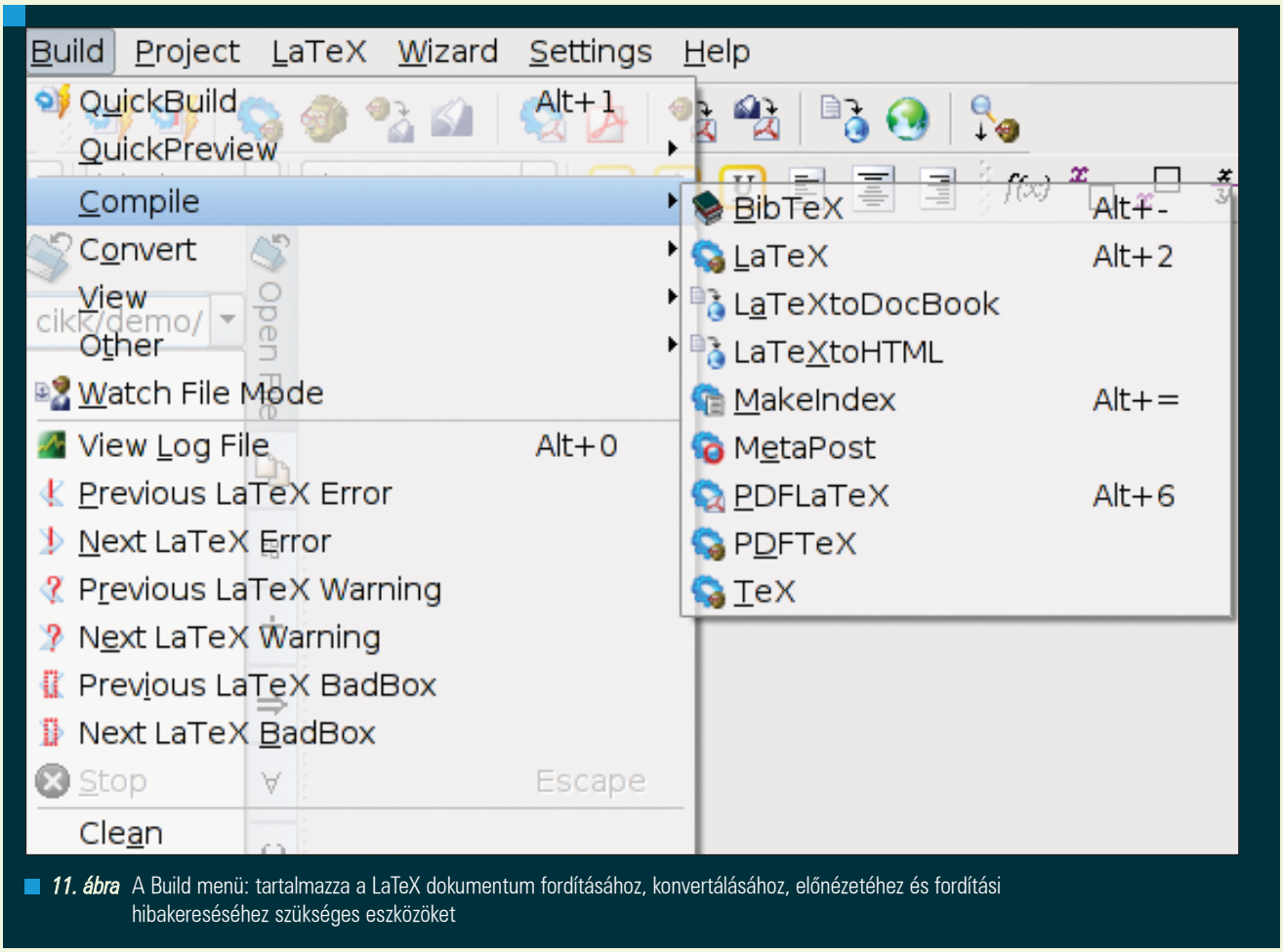
segítő eszközökben merül ki. *LaTeX* dokumentumok szerkesztéséhez a *Kile* menüsorának *LaTeX* nevű menüpontja is egyike a legfontosabb segítségeknek, amelyeket a *Kile* a szerkesztéshez nyújt. Ebben a menüben az összes fontosabb *LaTeX* környezeti és formai elem illetve stílus elérhető könnyen, gyorsan, ráadásul anélkül, hogy valaki fejből és pontosan ismerné az összes ilyen elem nevét és szintaxisát. Lehetőségünk van táblázatok, listák, felsorolások, matematikai környezetek, bibliográfiai elemek könnyű beszúrására, de a *LaTeX* menü alpontjai között találunk a dokumentum stílusának beállítására vonatkozó funkciókat (*Preamble* menüpont), fejetek és alfejezetek definiálását és beszúrását segítő funkciókat (*Sectioning*), de képek beszúrását könnyítő elemeket is (*Floating environment*).

A *Varázslók* (*Wizard*) menü elemei további hasznos funkciókat tartalmaznak, mint például táblázatok gyors definiálása és beszúrása.

A *Könyvjelzők* (*Bookmark*) menü lehetőséget ad arra, hogy bárhol a szerkesztett szövegben könyvjelzőket helyezzünk el (*Bookmark* menü *Set Bookmark* menüpontja vagy *CTRL+B*) amelyek között *Alt+PageUp* és *Alt+PageDown* billentyűkombinációkkal gyorsan mozoghatunk.

Mivel gyakorlatilag a *Kile* nem más, mint egy szövegszerkesztő, fontos kérdés az is, hogy van-e lehetőség helyesírás-ellenőrzésre. A válasz természetesen igen. A szabad forrású szoftverfejlesztés egyik előnye a modularitás, ami a *Kile* esetében is többszörösen megmutatkozik. Egyik jó példa erre a helyesírás-ellenőrzés. A *Kile* az *Ispell*, *Aspell* vagy *Hspell* helyesírás-ellenőrző alkalmazásokat is képes használni. Ezek kiválasztása és az ellenőrizendő nyelv beállítása a *Beállítások-Kile beállításai-Helyesírásellenőrzés* (*Settings-Configure Kile-Spelling*) menüpontban lehetséges. Ezután a dokumentum szerkesztése közben a helyesírás-ellenőrzés az *Eszközök* (*Tools*) menüből érhető el.

Szintén jól jöhet az üzenet panel (*Message Bar*, a *Kile* ablakának alsó részében, 1. ábra), amelynek három panelje a *Log&Messages*, *Output*, *Konsole* közül az első a dokumentum fordításakor, konvertálásakor és előnézeténél fellépő esetleges hibákról értesít, a második a fordítások/konvertálások kimeneteit tartalmazza, míg a harmadik egy teljes értékű parancssori konzolt szolgáltat a *Kile* felületén belül. Míg az első kettő információk közlésére szolgál, a konzol jól jöhet gyors fájl-műveletek elvégzéséhez,



11. ábra A Build menü: tartalmazza a LaTeX dokumentum fordításához, konvertálásához, előnézetéhez és fordítási hibakereséséhez szükséges eszközöket

amire úgy van lehetőségünk, hogy nem kell elhagynunk az alkalmazást és külön konzol-programot indítanunk.

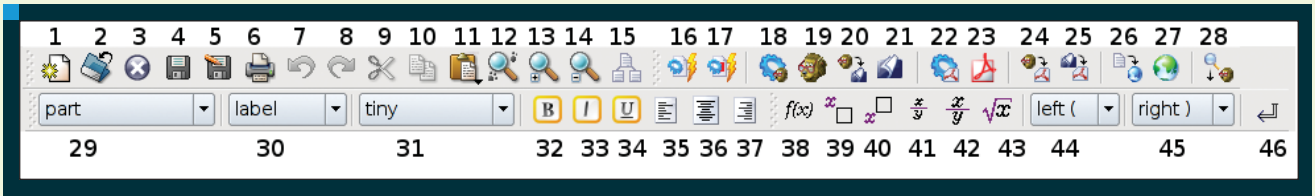
Dokumentumok fordítása, formátumok, előnézetek

Miután megismertük a *Kile* fontosabb részeit, szerkesztést segítő eszközeit, felületi elemeit, fordítsunk időt arra is, hogy a készülő *LaTeX* dokumentum-forrásunk lefordításához, konvertálásához és megtekintéséhez szükséges eszközöket is megismerjük. A *Kile*-ban mindezekre találunk könnyen használható eszközt. Kezdjük először a fordítással.

Ahhoz, hogy egy *LaTeX* utasításkészlettel felépített dokumentumból nyomdakész dokumentum váljék, szükséges egy úgynevezett fordítási lépés, amely a *LaTeX* utasításokkal teletűzdelt dokumentum-forrásunkból *DVI*, *PostScript* (*.PS*), *PDF*, vagy akár *HTML* formátumú dokumentumot készít. A *Kile* ilyen célú eszközeit a *Fordítás* (*Build*) menüben találhatjuk, ahogy azt a *11. ábra* is mutatja. A fordítási eszközök fontosabb elemei a *Kile* eszköztáraiból is gyorsan elérhetők, amelyek a *12. ábrán* láthatók és hamarosan az egyes elemeket be is mutatjuk.

Azt, hogy az egyes fordítási eszközök elindításakor pontosan mi is történjen, a *Beállítások-Kile beállítások-Fordítás* (*Settings-Configure Kile-Build*) menüpontban adhatjuk meg. A továbbiakban az alap-beállításokra hagyatkozunk, mivel szinte minden esetben ezek tökéletesen megfelelőek.

A *LaTeX* forrás fordításának első lépése egy *DVI* (*device independent; eszközfüggetlen*) formátumú dokumentum generálára, amely aztán bármely más formátumba könnyen konvertálható. A *DVI* fájl előállítása a *Build* menü *Compile* almenüjéből a *LaTeX* menüponttal lehetséges (vagy az *Alt+2* billentyűkombinációval). A fordítás esetleges hibáit a *Kile* alsó, *Message Bar* ablakában láthatjuk. Sikeres fordítás után a keletkezett *DVI* fájlt a *Build* menü *View* almenüjének *ViewDVI* menüpontjával nézhetjük meg (vagy az *Alt+3* billentyűkombinációval), amely egy a *Kile*-ba integrált dokumentumnéző-



12. ábra A Kile hasznos eszköztárai. A számokhoz tartozó leírásokat lásd a szövegben.

1. táblázat ***A Kile eszköztára*** *(lásd a 12. ábrát)*

1	Új dokumentum létrehozása
2	Létező dokumentum megnyitása
3	Aktuális dokumentum bezárása
4	Aktuális dokumentum mentése
5	Aktuális dokumentum mentése más néven
6	Aktuális dokumentum nyomtatása
7	Előző szerkesztési művelet visszavonása
8	Előző szerkesztési művelet ismétlése
9	Kijelölt tartalom kivágása a vágólapra
10	Kijelölt tartalom másolása a vágólapra
11	Vágólap tartalmának beillesztése
12	Szöveg keresése
13,14	Szerkesztő betűméretének növelése/csökkentése
15	Az aktuális projektben a dokumentum master-ként (fő dokumentum) beállítása
16	Gyors fordítás (a Settings-Configure-Build beállításai szerint)
17	Kijelölt szövegrész fordítása
18	DVI generálás
19	DVI előnézete
20	DVI-PS konverzió
21	PS előnézete
22/23	PDF generálás és előnézet
24/25	DVI-PDF és PS-PDF konverzió
26/27	TML generálás és előnézet
28	DVI generálása és ugrás a DVI fájlban arra a pozícióra ahol a forrásban voltunk
29	Fejezet/alfejezet/stb. Beszúrása
30	Index/lábjegyzet/referencia/stb. Beszúrása
31	Dokumentum fontméretének beállítása
32/33/34	Kijelölt szövegrész félkövér/dőlt/aláhúzott tulajdonságának beállítása
35/36/37	Szövegrész formátumozása: sorkizárt, jobbra/balra zárt
39-45	Matematikai képletek és zárójelezések egyszerű beszúrása
46	Új sor jel beszúrása a LaTeX dokumentumba

vel megmutatja a keletkezett dokumentumot. Az előállított ***DVI*** fájlt a hordozhatóság miatt ***PostScript*** (***.PS***) vagy ***PDF*** formátumba ajánlott (és sokszor kötelező) konvertálni, amire szintén lehetőséget ad a ***Kile***, a ***Build*** menü ***Convert*** almenüjében. Lehetőségünk van ***DVI-PS***, ***DVI-PDF*** és ***PS-PDF*** konvertálásokra, amelyeknek kimenetét szintén meg tudjuk tekinteni a ***Build*** menü ***View*** almenüjének segítségével (***ViewPS***, ***ViewPDF***, stb.).
Az egyes fordítási funkciók is természetesen elérhetők gyorsbillentyűkkel és a fontosabbak a ***Kile*** eszköztáraiból is, ahogy az a *12. ábrán* látható. Az *1. Táblázatban* tartalmazza az eszköztárak elemeinek rövid leírását. A számozás a *12. ábrának* fele meg.

Végszó

A ***Kile*** természetesen nem ad megoldást mindenre, nem egy olyan eszköz amivel mindenféle ***LaTeX*** tudás nélkül szerkeszthetnénk dokumentumokat, de nem is ez a fő célja. A ***Kile*** megpróbálja egyszerűbbé és hatékonyabbá tenni a ***LaTeX*** dokumentumok szerkesztését azzal, hogy eszközeivel gyorsítja a ***LaTeX*** parancsbevitelt, ezáltal csökkenti a ***LaTeX***-re fordított időt, így az szerkesztési időt arra fordíthatjuk ami fontosabb: a dokumentum tartalmára. A számos előny mellett természetesen találunk hiányosságokat, amelyek – mint mindig – nagyon szubjektívek is lehetnek. Például, ha a kevésbé ***LaTeX***-hozzáértőkre gondolunk, akkor hasznos lenne egy vizuális egyenletszerkesztő, amivel elkerülhető lenne a hosszadalmas ***LaTeX*** alapú egyenlet-szerkesztés. Ettől eltekintve, a dokumentumszerkesztés minden szegletének könnyítésére és gyorsítására találunk benne eszközöket.
Noha arra törekedtünk, hogy bemutassuk a ***Kile*** sokoldalúságát a ***LaTeX*** forrás-dokumentumok szerkesztésének segítésében, maradnak be nem mutatott funkciók és eszközök. Szerkesztés közben felmerülő kérdések esetén azonban akad segítség. A ***Kile*** belső helyfüggő segítségnyújtása (***Help*** menüben a ***Context help*** menüpont) gyors segítséget adhat egyes parancsokról és funkciókról. A ***Kile*** saját dokumentációja is bőven tartalmaz információt a ***Kile*** használatának minden aspektusáról, mindenkinek javasoljuk ennek tanulmányozását elakadás esetén. ***LaTeX***-el kapcsolatos segítséghez jó kiindulási pont lehet az, illetve egy jó internetes kereső. Végül, minden kedves olvasónak kellemes és hasznos időtöltést kívánunk a ***Kile***-al végzett ***LaTeX*** szerkesztésben.

Kovács Levente
(leventek@gmail.com)

26 éves informatikus- és villamosmérnök. Évek óta használ különféle Linux disztribúciókat. Fontosnak tartja a nyílt forrású szoftverek és fejlesztés előnyeinek megismertetését az emberekkel.

KAPCSOLÓDÓ CÍMEK

- http://kile.sourceforge.net/
- http://kile.sourceforge.net/Documentation/html/credits.html
- http://kde-cygwin.sourceforge.net/
- http://kile.sourceforge.net/Documentation/html/
- http://it.metr.ou.edu/latex/latex2e.html


Linuxvilág
www.linuxvilag.hu

Fájlba zárt szellem, MS Word dokumentumok kezelése Linux alatt

Mihez kezdjünk újonnan kapott, vagy régen elkészített MS Word dokumentumainkkal Linux alatt? Hogyan olvashatjuk el őket egyszerűen, a karakteres vagy akár a grafikus felületen?

Az alapvető probléma a *doc* formátumú szöveges állományokkal az, hogy a *doc* zárt fájlformátum, vagyis csak a készítők tudják, hogy pontosan hogyan is épül fel a fájl, milyen szimbólumok mit jelentenek, és ez alapján hogyan kell formázni a benne lévő szöveget.
Arról felesleges beszélni, hogy ez kinek jó, és kinek nem jó, mindenesetre az általam írt szöveges fájl tartalma a saját tulajdonom, melyhez bármikor, bárhol hozzá kellene férnem, lehetőségemnek kellene lenni tovább szerkeszteni, még akkor is ha, éppen nem rendelkezem érvényes *Microsoft Office* licencel.
Habár a *Microsoft* honlapján található egy ingyenes *doc* fájl megjelenítő, az sajnos nem támogatja a többi operációs rendszert, így *Linux* alatt sem fog futni. Esetleg valamilyen emulátor használatával még életre lehet kelteni, de a dokumentumot akkor is csak megnézni tudjuk vele, tovább szerkeszteni, módosítani nem.
Vegyük számba tehát, hogy milyen ingyenes és szabad programok állnak rendelkezésünkre, hogy elolvassuk újonnan kapott, vagy régi *doc* fájljainkat, illetve átkonvertáljuk őket valamilyen nyílt formátumra. De melyik a legjobb eszköz erre?
Az *OpenOffice*, vagy az *Abiword*, esetleg a *KOffice*? Vagy van még egyéb lehetőségünk is? Mit csináljunk akkor, ha csak egy karakteres terminál áll rendelkezésünkre? Egyszerű, olvassuk tovább a cikket!

Word dokumentumok olvasása a karakteres felületen

Amennyiben a dokumentum nem tartalmaz képet, vagy a kép számunkra nem fontos, akkor arra is lehetőségünk van, hogy a fájlt a karakteres felületen olvassuk el. Erre is több lehetőségünk van, nézzük meg először a *catdoc* programocskát, ez tulajdonképpen hasonló elven működik, mint a sokak által ismert *cat*, de a bemenete egy *doc* formátumú fájl, a kimenete pedig minden formázás nélküli szöveg. Sokszor a célnak ez is megfelel. Használata közel sem bonyolult, alapvetően két választási lehetőségünk van, aszerint, hogy a kimenet *ascii* vagy *tex* formátumú legyen.
Lássunk, egy példát az *ASCII* formátumú kimenetre:

```
catdoc -fascii valami.doc
```

ekkor a szöveg a terminálra íródik ki, természetesen át tudjuk irányítani a kimenetet fájlba is, például

```
catdoc -fascii valami.doc
➥> olvashato.txt.
```

A `catdoc` programocskával, és némi *Bash* szkripttel pedig akár az összes *doc* formátumú fájlunkból készíthetünk egy egyszerűen olvasható szöveges verziót. A program kimenete lehet *Latex* is, ekkor a *-t* opciót használjuk a `-fascii` helyett.
Lehetőségünk van a kimenet karakterkódolásának megváltoztatására is, ami alapértelmezett esetben *utf-8*. Az elérhető karakterkódolások listáját a

```
catdoc -l
```

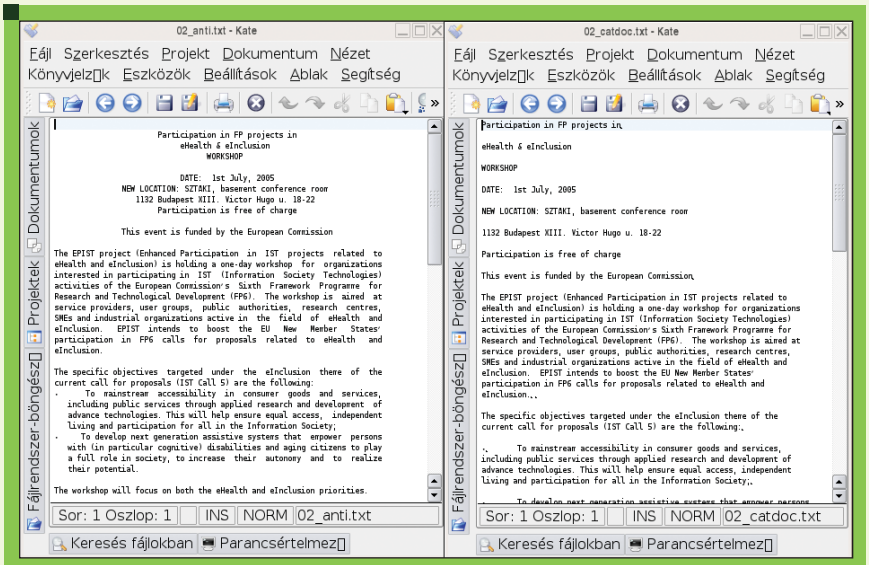
paranccsal kérdezhetjük le, és a `-d` opcióval használhatjuk, például:

```
catdoc -fascii -d8859-2
valami.doc.
```

A `catdoc` tényleg csak arra használható, hogy a fájlból kiolvassuk a szöveget. Amennyiben ennél valamivel nagyobb igényeink vannak, akkor próbáljuk ki az *antiword* programot. Az *antiword* már jóval kifinomultabb program, és még a karakteres felületen is láthatóan szebb eredményt produkál, mint az előző (*1. ábra*). Az eredeti szöveg sorkizárt, és az *antiword* a sorkizárást jól jeleníti meg a karakteres kimeneten, ellentétben az *catdoc* programmal. Különösen tetszett, hogy a sorkizárt szövegeket ténylegesen sorkizártként jelenítette meg a karakteres felületen is. Az *antiword* már használható *PostScript* fájlok készítésére is, ami teljes mértékben megfelel arra, hogy az eredeti *doc* tartalmát bárhol megjelenítsük, akár egy *PostScript* nyomtatónak utána átadjuk, de sajnos tovább szerkeszteni nem lehet. Az

```
antiword valami.doc
```

parancs szintén az alapértelmezett kimenetre rakja ki a `valami.doc` tartalmát *UTF-8* karakterkódolással, lehetőségünk van a megadni, hogy az elkészült szövegfájlban hány karakter lehet maximum egy sorban, a `-w X` opcióval, ahol `X` egy pozitív szám.



■ *1. ábra* Baloldalon látható az antiword és jobb oldalon a catdoc kimenete

Word dokumentumok konverziója

Az `antiword` programmal *PostScript* fájlt, a következő paranccsal készíthetünk:

```
antiword -p a4 -m 8859-2.txt
➥valami.doc > valami.ps
```

ahol `-p a4` jelenti, hogy *A4*-es oldalakra szeretnénk formázni a szöveget, `-m 8859-2.txt` pedig a *PostScript* fájl karakterkódolása, az *UTF-8* használata itt nem támogatott.
Mivel az `antiword` más formátumban történő exportálást nem támogat, ezért érdemes még tovább keresni hasonló programok iránt. Elég ígéretesnek tűnik a *wvWare* program, mely talán az eddigiek közül a legtöbb szolgáltatást nyújtja. Lehetőségünk van ugyanis, *HTML*, *PDF*, *Latex*, *ABW (Abiword)*, vagy akár *RTF* formátumra is átkonvertálni az eredeti *MS Word* fájlt.
A *antiworddel* ellentétben a *wvWare* a *PostScript* fájlokat nem egyből a *doc*-ból készíti el, hanem először *Latex* fájlt készít belőle, és abból készít *ps*, illetve *pdf* fájlokat. Így ha *ps* fájlt szeretnénk kimenetként kapni, akkor feltétlenül szükségünk lesz a *tetex* csomagra.
Bizonyos átalakításoknál a *wvWare* a fejlesztők szerint szebb kimenetet ad, amennyiben a *links*, *elinks*, és *lynx* programok fel vannak telepítve.
A *wvWare* képes *HTML* formátumba is konvertálni, apró csellel persze ezt is megtekinthetjük a karakteres terminálon, mégpedig úgy, hogy a *wvWare* kimenetét beleirányítjuk a *w3m* bemenetébe:

```
wvWare -x /usr/share/wv/
➥wvHtml.xml valami.doc | w3m
➥-T text/html
```

Előfordulhat, hogy nekünk máshol helyezkedik el az adott fájl. Ez a megoldás azért jobb, mint ha simán csak a `text` kimenetet választottuk volna, mert így a fájlban található linkekre azonnal rákattinthatunk. Az előbb a `-x` kapcsolót akár el is hagyhattuk volna, mert azt csak akkor kell megadni, ha nem az alapértelmezett *HTML* formátumba történik a konverzió.
A *HTML* formátum nagy előnye, hogy a képek nem a fájlban vannak, hanem csak a hivatkozás található meg rájuk, és mindegyik külön fájlban tárolódik. Ezért ha egy *doc* fájlból ki akarjuk nyerni az összes képet, akkor csak konvertáljuk át a *wvWare* segítségével *HTML* formátumúra, mellesleg a program támogatja azt is, hogy az egy *doc* fájlban lévő képek neve elé valamilyen megkülönböztető karaktersorozatot tegyünk. Ezáltal rögtön tudni fogjuk, hogy melyik kép melyik *HTML* fájlhoz tartozik.
Talán az egyik leghasznosabb konverzió, amit a *wvWare* tud, az a *Latexre* történő átalakítás, és rögtön kétféle *Latex* kódot is képes generálni, attól függően, hogy mennyi formázást kívánunk megtartani az eredeti fájlból.

Amennyiben az `-x` kapcsolót */usr/share/wv/wvCleanLaTeX.xml* paraméterrel hívjuk meg, akkor a kimeneti fájl csak minimális formázást tartalmaz, de ha a */usr/share/wv/wvLaTeX.xml* paraméterrel, akkor a kimeneti *Latex* fájl megpróbál minél jobban hasonlítani a *Word* által megformázott *doc*-ra. Ekkor azonban olyan formázások kerülnek bele a fájlba, melyeket utána kézzel elég nehéz lesz szerkeszteni.
Az egyszerűség kedvéért *wvWare* programhoz tartozó segédprogramokkal (*wv**) nem kell megkeresnünk az adott *xml* fájlt, hanem csak ki kell adnunk a

```
wvLatex valami.doc valami.tex
```

vagy

```
wvCleanLatex valami.doc
valami.tex
```

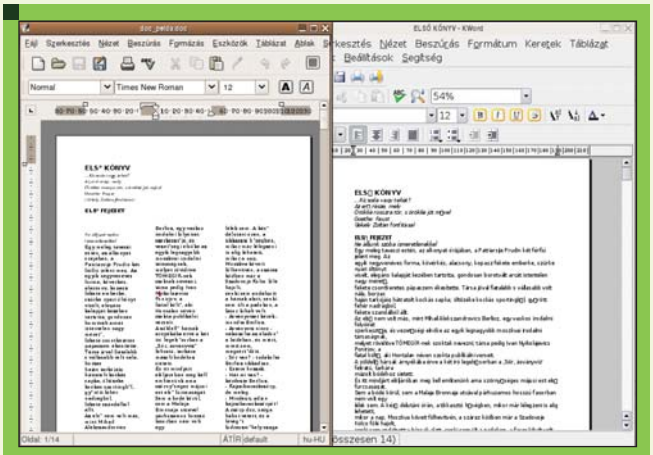
parancsot, és máris rendelkezésünkre áll a kész *Latex* kód.

Grafikus felületű programok

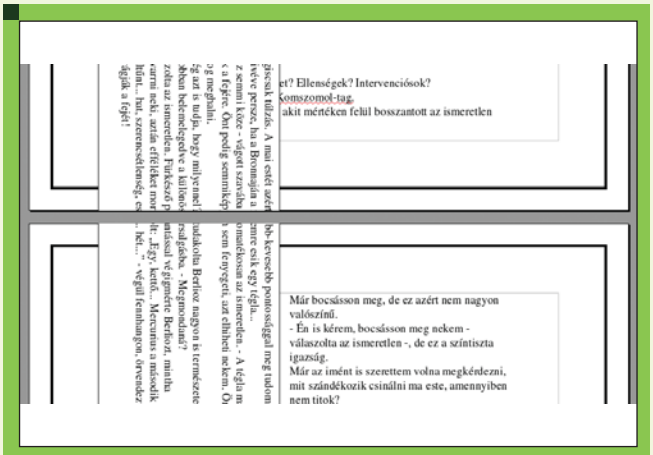
Miután megismerkedtünk a karakteres felületen keresztüli *MS Word* dokumentumok olvasásával, konverziójával, térjünk át egy sokkal izgalmasabb, és valószínűleg többet is használt területre. Ezek pedig a grafikus felületen futó programok. Bizonyára mindenki ismer ezek közül legalább egyet, de valószínűsítem, hogy sokan akár az összessel találkoztak már.
Most nézzük meg ezeket a szövegszerkesztőket abból a szempontból, hogy mennyire értik meg az *MS Worddel* készített fájlokat. A *KOffice* hátránya, hogy csak *Linux* alatti verzió van belőle, míg az *OpenOffice* és az *Abiword* rendelkezik *Windows* vagy *Macintosh* alatt is használható verziókkal.

Az Abiword

Talán erről a programról lehet hallani a legkevesebbet, az összes közül.
A többivel ellentétben az *Abiwordnek* csak szövegszerkesztő funkciói vannak, míg a *KOffice*, és az *OpenOffice* tartalmaz táblázatkezelő, és prezentáció készítő programokat is.
Az *Abiword* előnyére válik ugyanakkor, hogy nagyon kis méretű alkalmazás, a legutóbbi verzió telepíthető csomagja körülbelül 3,5 Mb. Ennek megfelelően kevesebb erőforrásra van



■ *2. ábra* Az Abiword egyszerűsége ellenére jobban megjeleníti az MS Word fájlokat a KWord-nél



■ *3. ábra* Az OpenOffice 1.x sem képes tökéletesen megjeleníteni mindent

szüksége, mint a másik két programnak, és a doc formátumú szöveges fájlokat is érezhetően gyorsabban nyitja meg. A konverzió még nagy dokumentumok esetén is pillanatok alatt megtörténik. Ebből természetesen nem csak azért van, mert a program kicsi, hanem mert az ***Abiword*** fájlszűrője nem tökéletes, és bizonyos formázásokat figyelmen kívül hagy.
Az oldalszegélyek egyáltalán nem jelennek meg, ahogyan néha a képek sem a dokumentumban, a ***Wordart*** szövegek pedig egyáltalán nem támogatottak. A program jól kezeli ellenben a karakter formázásokat, a táblázatokat, a hasábokat, és a címjegyzékkel is megbirkózik, habár az utóbbi esetében az egyes al-al fejezetek számozásánál oda nem illő karakterek jelentek meg.
Az ***Abiword*** nem támogatja a hierarchikus felsorolást, ezért a konverziónál bár a behúzások megmaradtak, de a kis piktogrammok már nem tesznek különbséget az egyes szintek között.
Az ***MS Word*** dokumentumok beolvasása során az ***Abiword*** jól megjelenítette a tabulátorokat, de a tabulátorok közötti kitöltést nem. A lábjegyzet, élőfej, élőláb szintén támogatott a programban, de az élőfejben, élőlábban lévő dátum, és idő mezők többször jelentek meg egymás után egy oldalon, így a felesleges mezőket kézzel kell kitörölnünk.
Az ***Abiwordben*** továbbá egy oldalon nem lehetnek különböző oszlopszámú hasábok sem.
Ne felejtsük el ugyanakkor, hogy egy programot nem feltétlenül az határoz meg, hogy milyen hűen kezeli az ***MS Word*** dokumentumokat. Egyszerűbb szöveges állományok megnyitására azonban jól használható, platform független szövegszerkesztő.

A KOffice

A ***KOffice*** programcsomag szövegszerkesztője a ***KWord*** program, ahogyan már ezt megelőzően említettem a ***KOfficenak*** nem létezik ***Windowsos***, vagy ***Mac OS X*** verziója, ellenben a ***Fink*** projekt (➲ ***http://fink.sourceforge.net***) keretében elérhető a ***Mac OS X*** verzió.
A ***KOffice*** jól együttműködik a ***KDE*** ablakkezelő rendszerrel, sőt szüksége is van a ***kdebase*** csomagra, és a program felülete a szabványos ***KDE*** programokhoz illeszkedik.
Sok különböző fájlon végzett konverziók után inkább csalódott voltam a programmal kapcsolatban, mintsem megelégedett. Szignifikánsabb jobb eredményt nem sikerült vele előállítani, mint amit az ***Abiworddel*** is sikerült. Ami nekem nagyon hiányzott, az a hasábok támogatása volt. Tulajdonképpen a táblázatok, szövegformázások, felsorolások kezelése jól meg volt oldva a programban. Az ***MS Word*** dokumentumba beágyazott grafikákat sem jelenítette meg, ami kissé érthetetlen, hiszen a ***Kivio*** is a ***KOffice*** része, ami egy ***MS Visio*** klón, ezért jogosan vártam volna el, hogy ilyen téren maga mögé utasítsa az ***Abiword*** programot.
Az ***Abiworddel*** ellentétben a ***Kword*** jól kezeli a tabulátorokat, és az előtte való kitöltést is képes megjeleníteni.
Az élőfej, és élőláb rendesen működik a szövegszerkesztőben, illetve a dátum, oldalszám, és idő mezők is jól jelennek meg.
Durva hibának találtam azt, hogy a szövegdobozban lévő szöveget a ***KWord*** képtelen volt kiolvasni a fájlból, ezért az gyakorlatilag elveszett. Míg az ***Abiword*** legalább kiolvasta és a dokumentum végére fűzte a szövegdoboz tartalmát, habár az a szöveg közepén helyezkedett el eredetileg.

Az OpenOffice.org 1.x

Minden bizonnyal az összes eddig megvizsgált program közül a legismertebb irodai alkalmazás, a legtöbb funkcionalitással, és természetesen az ezzel együtt járó kicsit lassabb működéssel. A szoftver fejlesztésekor kifejezett hangsúlyt fektettek arra, hogy amennyire csak lehet kompatibilis legyen a ***Microsoft*** hasonló célokra készült termékével, amely érezhető is a használat során.
Tulajdonképpen ez az egyetlen alkalmazás, amely szinte mindennel megbirkózott. A dokumentumba beszúrt összes képet megjelenítette, ügyelve arra, hogy a kép tulajdonságait is megtartsa, attól függően, hogy a kép a szöveg előtt, vagy mögött helyezkedett-e el. Az egyetlen probléma talán az volt, hogy a beágyazott grafikákat tovább nem lehetett szerkeszteni, mert az egy dobozba tartozó összes objektum egy képként jelent meg az ***OpenOffice.org***-ban.
Az ***OpenOffice.org*** még olyan kevésbé fontos részleteket is precízen jelenített meg, mint az oldal szegélyek, a szövegdobozban a szöveg irányultsága, a ***WordArt***-tal készített feliratok, és tulajdonképpen még sorolhatnám azon formázási elemeket, melyeket sikerrel átvett a ***doc*** formátumból.

1. táblázat *Konklúzió*

Funkció	Abiword	KWord	OO.o 1.x	OO.o 2.x
Karakter formázások	✓✓	✓✓	✓✓	✓✓
Hasábok	✓	✗✗	✓✓	✓✓
Címjegyzék	✓✗	✓✓	✓✓	✓✓
Beszúrt képek	✓✗	✗✗	✓	✓✓
Tabulátorok	✓✗	✓✓	✓✓	✓✓
Felsorolás, számozás	✓	✓✓	✓✓	✓✓
Táblázatok kezelése	✓	✓	✓✓	✓✓
Szövegdoboz	✗	✗✗	✓	✓✓
Élőfej, élőláb (oldalszám, dátum mező)	✓	✓✓	✓✓	✓✓
Rajzok kezelése	✗✗	✗✗	✓	✓
WordArt	✗✗	✗✗	✓	✓✓
Oldalszegélyek	✗✗	✗✗	✓	✓
Lábjegyzet	✓✓	✓✓	✓✓	✓✓
Összhatás	✓	✗	✓✓	✓✓

Magyarázat: ✓✓ nagyon jól importál; ✓✗ bizonyos esetekben jól importál; ✗✗ egyáltalán nem támogatott; ✓ jól importál; ✗ inkább rosszul importál

Kisebb problémák itt is voltak a megjelenítéssel, például a szövegdoboz (*3. ábra*) két oldal közé került, és kézzel kellett áthelyezni egy olyan helyre, hogy az oldal határain belülre kerüljön.

Az OpenOffice.org 2.0 beta

Mivel az ***OpenOffice.org 2.0*** nagyon sok újítást tartalmaz az előző verzióhoz képest, de a cikk írásának pillanatában nincsen belőle stabil verzió, ezért úgy gondoltam, hogy jobb, ha mindkét változatot külön-külön veszem górcső alá.
Az előző verzió összes jó tulajdonsága tulajdonképpen igaz a 2.0-ra, azért csak azokat a területeket említem meg, ahol a változás volt tapasztalható.
Első ránézésre nagyon hasonló volt az összes megjelenített dokumentum, és csak egyetlen hiányosságot találtam benne. Viszont egy kisebb beállítás után az ***MS Word*** és ***OpenOffice 2.0*** alatt pontosan ugyanúgy nézett ki a dokumentum.
Ez a hiányosság mégpedig az oldalszegélyek kapcsán került bele a dokumentumban, ugyanis az ***OpenOffice*** és az ***MS Word*** alapértelmezésként másképpen állítja be az oldalszegély helyzetét. Míg az ***OpenOffice*** alapbeállításként a szegélyt szorosan a betűk mellé helyezi, addig az ***MS Word*** a szegély és a betűk között hagy némi távolságot. Ettől a kis különbségtől a megnyitás után egy kicsit másképpen néz ki a dokumentum, de átállítás után teljesen azonos kép fogad minket.
A fejlesztők a szoftver esetében még arra is figyeltek, hogy a címjegyzékben lévő hivatkozások is működjenek. Tulajdonképpen itt sorolhatnám azokat az apró finomításokat, amit az ***OpenOffice 2.0***-át tökéletesítik az elődjéhez képest, de minden apró részletre hely hiányában nem térnék ki.

Konklúzió

A táblázatban megpróbáltam összefoglalni, körülbelül 20 ***Word*** dokumentum konvertálása közben történt tapasztalataimat. A ***Word*** dokumentumok között vegyesen voltak különböző verziójú ***Word***del készített fájlok. Ez a szám tulajdonképpen kevés arra, hogy azt merjem állítani, hogy bizonyos funkciók minden esetben működnek, úgyhogy a valós életben történt felhasználás során eltérések természetesen lehetségesek.

Mivel tulajdonképpen a ***doc*** formátum egy fekete doboz, és a fejlesztők mindent megtettek annak érdekében, hogy a belső struktúráját visszafejteni a lehető legnehezebb legyen, ezért elképzelhető az, hogy egyes esetekben mégsem tudjuk jól megjeleníteni az adott fájlt. Mellesleg ez sokszor még a Microsoft termékek egyes verziói közötti váltás során is problémákat okozott, hogyan várhatnánk el akkor, hogy a tulajdonképpen a sötétben tapogatózó fejlesztők ennél jobbat alkossanak?!
Mindenkinek azt ajánlom, aki ***doc*** fájlokkal foglalkozik, hogy mindenképpen szerezze be az ***OpenOffice 2.0*** béta verzióját, mert sokkal megbízhatóbban kezeli az ***MS Word*** formátumát mint az előző verzió. Sajnálatos módon a két verzió nem fér meg egymás mellett, így nem lehet párhuzamos használni mindkettőt. Ezért amennyiben a 2.0 nem tűnik elég megbízhatónak számunkra ahhoz, hogy mindenhol lecseréljük vele az ***OpenOffice 1.x***-et, legalább néhány helyen érdemes ezt megtenni, főleg ott, ahol sok ***doc*** formátumú bejövő anyagra kell számítanunk.

Az optimista jövő

Amennyiben hihetünk az Interneten terjengő híreszteléseknek, és a piacot uraló termék fejlesztőinek, akkor a következő irodai programcsomag már olyan fájlformátumot fog használni, amelyik teljesen nyílt lesz. Ez nem csak azért nagyszerű, mert ezentúl ezzel a termékkel készült fájlokat gond nélkül lehet majd kezelni ***OpenOffice*** vagy akár ***Abiword*** alatt, hanem azért mert végleg megoldódnak a konverziós problémáink, és az összes eddigi ***doc*** formátumban tárolt adatunkat könnyedén konvertálhatjuk át ebbe az új, nyílt formátumba.

Horváth Ernő
ernohorvath@gmail.com
24 éves, műszaki informatikus. Három évvel ezelőtt ismerkedett meg komolyabban a Linux rendszerekkel és emellett érdeklődik még a robotika és a biztonságtechnika iránt is. Ha lenne szabadideje sokat kirándulna, biciklizne és filmeket nézne.

Diagramok könnyen, gyorsan – Dia

Végre egy program, amit nem kell megtanulni használni – annyira egyszerű. A Dia egy GTK alapú diagramszerkesztő program. Nem vállal ennél többet, viszont tökéletesen megfelel arra a célra, amire szánták.

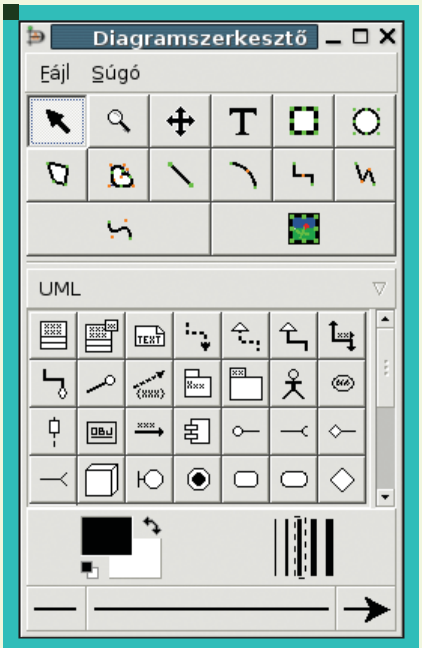
Azért mielőtt dicshimnuszokat zengenénk a készítők szerénységéről, megjegyzem, hogy a legtöbb dokumentáció a *Microsoft Visio* alternatívájaként vezeti be az alkalmazást. Ebben a témában nem szeretném az érvek és ellenérvek végeláthatatlan sorával untatni az Olvasót. Egy biztos: a *Dia*-t ki kell próbálni!

Telepítés

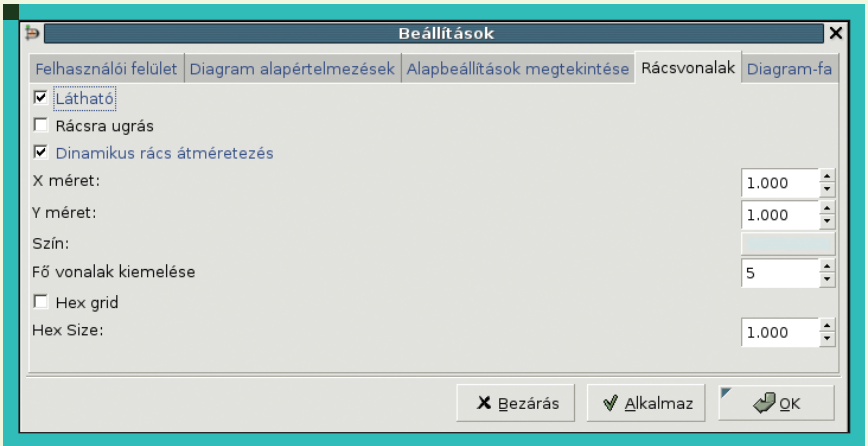
Deb, illetve *RPM* alapú rendszereken biztosan megtalálható az előfordított csomag, melyet csupán telepíteni kell. Ha véletlenül nem állna rendelkezésre csomagkezelőnkön keresztül a legfrissebb *Dia*, a hivatalos oldalról indulva, a ➲ *http://www.gnome.org/projects/dia/* címről letölthető a forráskód. Jelenleg a 0.94-es a legfrissebb stabil változat. Ennek fordításához a *GTK* függvénykönyvtár fejlesztői csomagjára is szükség van.

Első találkozás

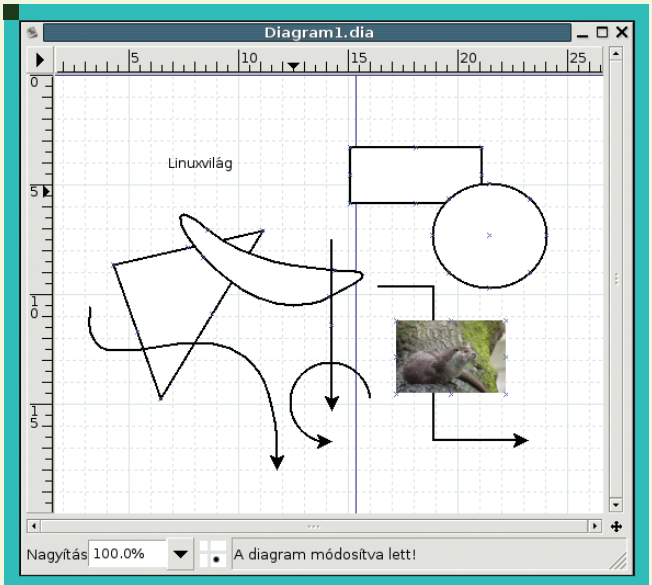
Indítás után az *1. ábrán* látható ablak tárul elénk. A kezelőfelület nagyban hasonlít a *Gimp*-ben megszokottra. Egy fő ablakban érhetők el a legfontosabb, általános műveletek, minden egyebet a külön ablakban szerkesztett ábrán a jobb egérgombbal elérhető helyi menüben találunk. A fő ablak Fájl menüjében található egy Tulajdonságok... menüpont, amire érdemes egy rövid pillantást vetni.
Itt érdemes ellenőrizni, hogy a rácsvonalak láthatóak-e, illetve ízlésünknek megfelelően be- és kikapcsolhatjuk a rácsra ugrást. Ezt követően, ha a program indításával még nem jelent meg egy új diagram, a *Fájl* menü *Új* pontjának kiválasztásával hozzunk létre egyet. Most pedig lássuk, milyen eszközök találhatók a fő ablak felső részében.
A kis fekete nyíl a meglévő objektumok módosítására ad lehetőséget. Mint azt látni fogjuk, a rajzterületen elhelyezett objektumok különféle kapcsolódási pontokat kínálnak. Ezt az eszközt kiválasztva ezek a pontok megragadhatók, és az objektum egy adott tulajdonsága így egyszerűen módosítható. Például egy téglalap egyik sarkát megragadva változtathatjuk a szélességét és a magasságát.
A nagyító mindannyiunk legnagyobb meglepetésére nagyításra használható. A bal egérgombbal kattintva közelít, míg ha a *Shift* billentyűt lenyomva kattintunk, távolít. A nagyítás



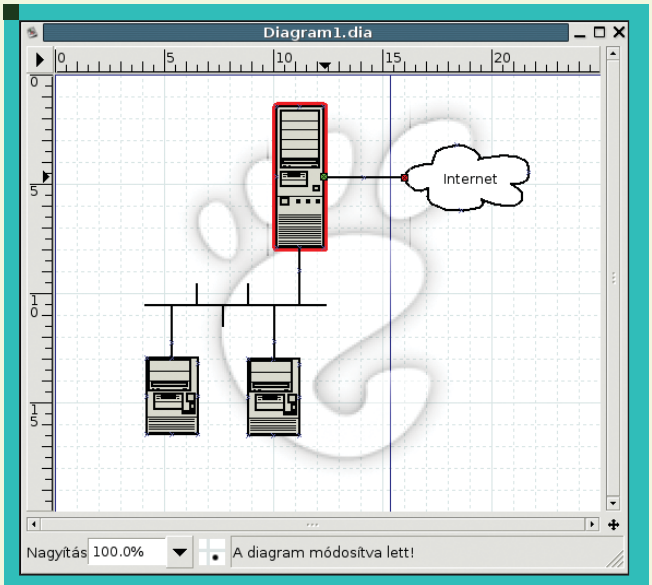
1. ábra A fő ablak



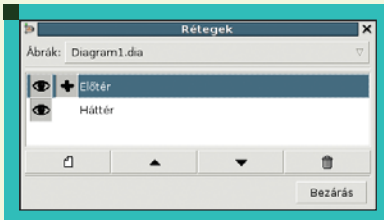
2. ábra A Tulajdonságok... ablak



3. ábra Egyszerű rajzolóeszközök



5. ábra Ragasszuk az Internetet az átjáróhoz...



4. ábra Rétegek kezelése

mértéke 5% és 2500% között változhat. Ugyanezt a diagram bal alsó sarkában látható szövegmező módosításával is elérhetjük.
A négy irányba mutató nyilakkal jelzett eszköz egy diagram könnyebb böngészését teszi lehetővé. Legjobban ahhoz hasonlítható, amikor egy szöveges képernyőn lenyomjuk a ***Scroll Lock*** gombot.
A további eszközök egyszerűbb rajzolóműveletek végrehajtásához használhatók. Ezek sorrendben a következők: szöveg, téglalap, ellipszis, sokszög, ***Bezier*** görbékkel határolt síkidom, vonal, körív, cikkcakkvonal, töröttvonal, Bezier-görbe és kép beszúrása. Ezek eredményét mutatja be a *3. kép*.
Az ablak alsó felében már összetettebb objektumok használatára nyílik lehetőség. Ezek különböző kategóriákba vannak sorolva aszerint, hogy milyen tudományos területen elterjedt jelölésrendszer eleméről van szó.
Az *1. ábrán* látható főablakban a kategóriát kijelölő legördülő menüből az ***UML***-t választottam, ezért alatta többek között osztályok, objektumok, szereplők, eseményforrások alakzatai találhatók. Ezeknek többsége temérdek beállítási lehetőséggel rendelkezik, melyeket legegyszerűbben dupla kattintással érhetünk el.
Az összetett objektumok alatt az előtér- és háttérszínt, a vonalvastagságot és néhány, vonallal kapcsolatos stílust állíthatunk be.

Közelebbről megvizsgálva

Hozzunk létre egy új diagramot ***Dia***-ban, és látni fogjuk, mennyire magától értetődő a használata.
Feladatunk egy egyszerű hálózat vázlatának elkészítése, egy ***Gnome*** tappanccsal a háttérben. Elsőként helyezzük el a tappancsot egy új diagramon. Mivel ez csak egy háttér lesz, érdemes a további munkát egy új rétegen folytatni, hogy véletlenül se mozdítsuk el, vagy méretezzük át a tappancsot. A jobb kattintással elérhető helyi menüből válasszuk ki a ***Diagram/Rétegek...*** menüpontot.
A *4. ábrán* látható ablakban hozzuk létre az új réteget, és nevezzük is el. A háttér módosíthatóságát kapcsoljuk ki a + jelre történő kattintással.
Ezután a főablakban válasszuk ki a Hálózatok kategóriát, és helyezzünk el egy ***Ethernet*** sínt a diagramon. Pakoljunk ki továbbá néhány ***PC*** mini tornyot, az ügyfél számítógépek ábrázolásához. Ezeket egy-egy egyszerű vonallal kössük hozzá a sínhez. Észrevehetjük, hogy a vonal ragadós, tehát az összeköttetés létrehozása után a mini tornyok mozgatásával a vonal is mozog.
Tegyünk még egy ***PC*** nagy tornyot átjáróként az ábrára, és kössük össze egy felhővel. A felhőbe írjuk bele, hogy Internet. Az *5. ábrán* látható fotó az utolsó simítások elvégzése közben készült.
A kész munkát dia formátumban menthetjük el a helyi menü Fájl/Mentés parancsával. Ez egy ***XML*** alapú leírása a felhasznált objektumoknak. Exportálhatjuk is a képet, többek között ***EPS***, ***PNG***, illetve ***SVG*** formátumokban. Ne feledkezzünk meg viszont arról, hogy ezek a képek már nem szerkeszthetők a programmal, ezért mindig tartsuk meg a dia állományt is.
Remélem sikerült felkeltenem azok figyelmét, akik még nem vették rá magukat, hogy kipróbálják ezt a remek programot.
Jó rajzolgatást!

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

IPCop
a hálózsaru

Szolgál és véd, ráadásul megoszt és felügyel is. Könnyen és gyorsan telepíthető – jó, jó, tudom, most majdnem mindegyik disztribúció ezt hangoztatja.. – speciális Linux változat az IPCop, s legfőbb célja tűzfalként védeni az internet felé és felől irányuló forgalmat, valamint igény esetén útválasztóként is tud üzemelni. Bővíthetősége kiváló, s csekély hardverigényéhez viszonyított rugalmas paraméterezhetősége miatt jó alternatívája lehet egy hardveres tűzfal/router eszköznek.

A tűzfalépítés alapanyagai

Hozzávalók: 1 db számítógép 2 db hálókártyával, 1 db *IPCop Linux*, 1 db eszköz az internet hozzáféréshez (például modem), 1 db switch és egyenes patchkábelek ha több gépet akarunk csatlakoztatni (ha csak egyet, akkor elég egy crosslink kábel, és a switch sem szükséges).
Végy egy számítógépet, de még jobb, ha leporolod a sarokban porosodó, elavultnak hitt konfigurációdat. Már egy *Pentium 1* kategóriájú (türelmesebbeknek akár 486-os is) is megteszi, de az se keseredjen el, akinek „csak” egy alfaprocesszoros gépe pihen használaton kívül, ezen is fut a szerzők szerint.
Ha kicsit többre vágyunk, például betörésdetektáló szolgáltatásra, akkor az ajánlott minimum körülbelül egy *Pentium II* tudású gép, legalább 64 MB RAM-mal, és körülbelül 1 GB merevlemezzel. Internetkapcsolatunktól függ, de általában két hálózati kártya elegendő ahhoz, hogy a belső hálózat felé meg tudjunk osztani például egy *ADSL* vagy kábeles kapcsolatot. Ez a roppant olcsó konfiguráció az árához képest sok segítséget tud nyújtani nekünk, néhány ezek közül:

- Tűzfal
- Útválasztó (router)
- Betörésdetektálás
- *DHCP* szerver (és kliens)
- Proxyszerver
- *DNS* gyorsítótár
- *VPN*
- forgalomtervezés, illetve *QoS*

Ne ijedjünk meg, ha nem mindegyik fogalommal vagyunk tisztában, ezek egy részét utólag lehet hozzáadni rendszerünkhöz, s kezdetben célunk pusztán egy internetmegosztásra is képes, védett kapcsolatot biztosító *Linux* üzembe helyezése.

Még két dolog szükséges a kapcsolathoz: az internetszolgáltatás eléréséhez tartozó hardvereszköz (például *ISDN*-kártya, kábelmodem stb.), valamint maga az *IPCop*. Utóbbit a ➲ *http://www.ipcop.org/* címről kiindulva tölthetjük le, körülbelül 40 megabájt lemezre mentésével. Ügyeljünk arra, hogy éles használatra ne bétaverziót szerezzünk be, hanem valamely újabb stabil változatot, például az 1.4.10-et.

Alapelvek

Az *IPCop* négy elkülönülő részre osztja a hálózati kapcsolatokat, tipikusan:

- *GREEN* interfész: általában a védendő saját belső hálózatunkat (ami lehet akár egy darab számítógép is) jelenti
- *RED* interfész: általában ő az internetkapcsolat, azaz a biztonsági szempontból legmegbízhatatlanabb kapcsolat

- *BLUE* interfész: az egyre népszerűbb vezeték nélküli (*WI-FI*) kapcsolatokat lehet bevonni a felügyelt rendszerbe
- *ORANGE* interfész: az ún. demilitarizált zóna (*DMZ*) eszközeit tartalmazza, azaz olyan szervereket és szolgáltatásokat, melyek internet felől is láthatók részben (például a honlapunkat üzemeltető saját webszerver, vagy a levelezőszerver), de védelmet igényelnek

Ezek közül minimálisan a *RED* és a *GREEN* konfigurálása szükséges ahhoz, hogy a védendő belső hálózat gépei kijussanak az internetre. Egy tipikus, kisebb irodai hálózat igényeit kielégíteni képes *DSL* vagy kábeles elérés esetén két hálózati kártya kell a gépbe: az egyik a *GREEN* interfész lesz, és a belső hálózat felé biztosítja az internetet, míg a másik a *RED* interfész, mely közvetlenül a szolgáltató által biztosított hardvereszközhöz csatlakozik (kábelmodem vagy *ADSL*-modem).
Ahhoz, hogy a tűzfal valóban védjen is, külön „alhálózatokat" (lásd később) kell képeznünk a különböző interfészekhez, és az internettel közvetlen kapcsolatban álló *RED* eszköz lesz az átjáró a belső hálózat gépei és az internetet biztosító szolgáltató fogadógépe között.

Praktikák

A tűzfalgép valójában egy 24 órás működésre tervezett szerverként kell funkcionáljon, ezért érdemes úgy beállítani, hogy a leginkább hibatűrő módon tudja feladatát ellátni. Ha lehetőség van rá, kapcsoljuk ki az energiatakarékos funkciókat a *BIOS*-ban. Állítsuk be, hogy automatikusan újrainduljon a gép – így például egy áramszünet után jó eséllyel magához tér külső beavatkozás nélkül is. Ha csak távolról szeretnénk adminisztrálni, akkor a billentyűzet, egér és monitor is eltávolítható, de ezeket is be kell állítani a *BIOS*-ban, nehogy hiánya miatt hibaüzenettel megálljon indításkor maga a számítógép. Telepítés után mindenképp javasolt a floppyról indítás lehetőségét megszüntetni, ugyanis biztonsági mentéseket írhatunk floppylemezre, s elég kellemetlen, ha egy bent felejtett lemez miatt nem indul el legközelebb a számítógép.
Azokat az eszközöket is nyugodtan letilthatjuk, melyekre nincs szükség – például hangkártya, *USB* portok – általában ezek nélkül vígan elvan a rendszer. Ugyanez igaz szoftveresen is, bár az *IPCop* disztribúciók többnyire eleve nélkülözik a szükségtelen és biztonsági szempontból kifogásolható csomagokat, viszont utólag különböző kiegészítők hozzáadásával saját magunk sokat ronthatunk az egyébként jól kialakított rendszer megbízhatóságán.
Elszántabbak akár teljesen csendes verzióját is használhatják, hiszen a dokumentációban olvasható egy leírás a merevlemez helyett *CF* memóriakártyáról futni képes *IPCopról*. Azonban ebben az esetben ne akarjuk az eredeti felépítésben szereplő verziót futtatni, hiszen a naplófájlok, a */var*, */tmp* stb. könyvtárak sűrű írási műveleteket igényelnek, ez pedig hamar tönkreteheti a flashmemóriát. Nézzük meg inkább a leírásban említett héjprogramot, és olvassuk el a szükséges segítséget a használathoz, hogy a fizikai memória közbeiktatásával elkerülhetőek legyenek ezek a problémák.

(Tűz)falazás előtti tudnivalók

Egy alapfunkciókkal már jól működő *IPCop* feltelepítése fájdalommentes és gyors az esetek többségében, még akkor is, ha nincs nagy tapasztalatunk *Linuxszal*. Ehhez mindössze a következő információkat kell előre megtudnunk:

- Állandó *IP* címet kaptunk-e az internetszolgáltatónktól, és ha igen, mi ez a cím, illetve mi a *DNS*-szerver(ek) *IP* címe. Általában ha ezt nem igényeltük külön, akkor változó címet kapunk, ekkor viszont a szolgáltató dinamikus *IP*-címeket kiosztó szerverének (*DHCP*) címére lesz majd szükségünk a beállításokhoz.
- Milyen típusú elérésünk van? Az *IPCop* szempontjából a következő négy protokoll és technikai megoldás közül kell választanunk majd: *PPoE*, *PPTP*, *DHCP*-vel kiosztott cím, vagy statikus *IP* cím.
- A hálózati kártyák típusa, ami elsődlegesen a rajtuk lévő chip fajtáját (például *Realtek RTL8169*), s nem a kártya gyártói fantázianevét (például *Ovislink GE-2032R*) jelenti.

Ha működő gépről kell megszerezni ezeket az adatokat, akkor a következő módon juthatunk egyszerűen ezekhez az információkhoz:
Linux használatakor a következő parancsot írjuk be:

```
dmesg | grep ^eth[0123]\:
```

És az `eth0: registered` as kezdetű sorokban szereplő információra figyeljünk, az `eth0`, `eth1` stb. jelentheti az egyes hálózati kártyákat.
Windows esetében pedig parancssorban adjuk ki a következő utasítást:

```
ipconfig /all > c:\halochip.txt
```

s a C meghajtó gyökerében keletkező `halochip.txt` fájlt megnyitva a „Leírás:…" mellett szereplő információkat írjuk fel magunknak.

15 perc a falazási szintidő – ráadásul még kőműves sem kell hozzá

A telepítés legegyszerűbb módja, ha a letöltött *CD*-képet felírjuk (természetesen a *CD*-kép lemezre írása opciót kiválasztva az adott íróprogramban), majd *BIOS*-ban beállítjuk, hogy a rendszerindítás *CD-ROM*-ról történjen, és behelyezzük a lemezt. Ha minden jól megy, hamarosan egy rövid karakteres üdvözlőképernyőt látunk, ahol a boot: melletti sorba írhatunk indítási opciókat. Ezek a következők lehetnek:

- *nousb* – *USB* eszközöktől mentes használathoz
- *nopcmcia* – ha nincs *PCMCIA* eszközünk, és általában asztali gépeken ez a helyzet;
- *dma* – elsősorban a *SIS* chipeket tartalmazó alaplapokhoz készült, a merevlemezkezeléshez lehet rá szükség

A következő képernyőn már a telepítés nyelvét tudakolja tőlünk a rendszer: válasszuk ki bátran a *Magyart*. Nemsokára megkérdezi, hogy *CD-ROM*-ról, avagy hálózatról

(*HTTP/FTP*) szeretnénk-e folytatni a telepítést. Utóbbi a gyakorlatban azt jelenti, hogy előre felmásoljuk a fájlokat például saját webszerverünkre, s az elérési útját megadva folytatjuk majd az installálást. Mi a könnyebbség okán maradjunk az előbbinél.
Ezek után sem törekszik a rendszer a kelleténél nagyobb interaktivitásra: közli, hogy elvégzi a particionálást, és létrehozza a szükséges fájlrendszereket. Nyugodtan hagyjuk rá, de legyünk tudatában annak, hogy ezzel a merevlemez korábbi tartalma eltűnik, átadva helyét a tűzfalszerver fájljainak. Ha valaki kíváncsiskodni szeretne, az *ALT* és az *F2* billentyűk együttes lenyomásával megtekintheti a részletes történéseket, az *ALT* és az *F12* együttes leütésével pedig leginkább a rendszerindításkor hibásnak jelzett modulokkal kapcsolatban láthat információt; visszaváltani a telepítő képernyőre az *ALT* és az *F1* nyomva tartásával lehet.
A telepítés következő fázisában visszatölthetnénk egy már korábban elmentett *IPCop* konfigurációt, azaz ha valamiért újratelepítjük a rendszert, esetleg ilyen módon frissítjük korábbi verzióról, akkor lehetőségünk van a már bevált beállításokkal használni a frissen életre keltett rendszert. Ezt mi most kihagyjuk, viszont a következő, Hálózat beállítása című párbeszédablak már fontos állomás a telepítésben. Ha a gépre próbáljuk bízni a *GREEN* interfészhez tartozó hálózati kártya felismerését, akkor válasszuk a Próba feliratú gombot, s ő jó eséllyel megtalálja az első hálózati kártyát (linuxosan szólva az *eth0* –t). Ha ez nem sikerülne, nekünk kell a Kiválaszt gombot megnyomni, s megtalálni a kártyát

1. ábra A RED interfész telepítése sem ördöngősség

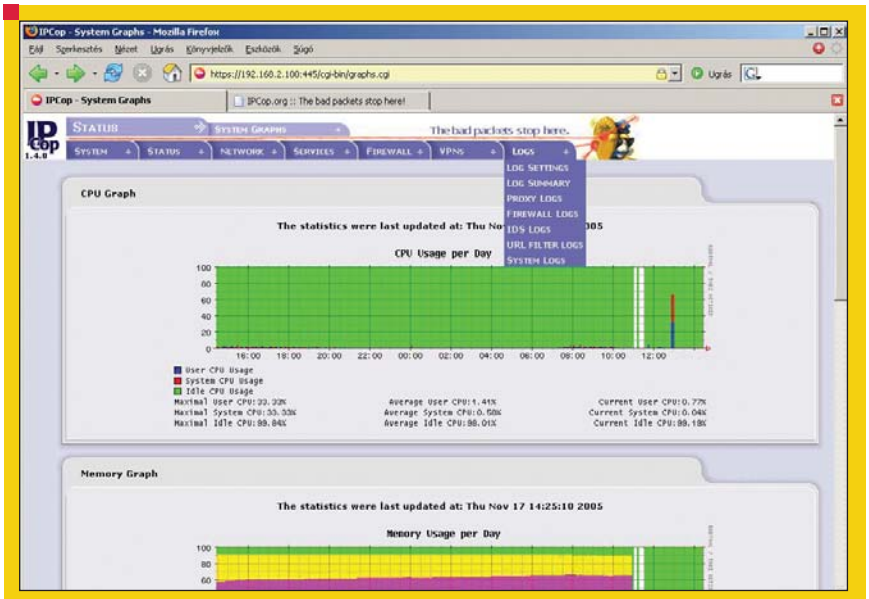
a listában, legrosszabb esetben a pontos paraméterezések ismeretében manuálisan beállítani.
Ha sikerült felismertetni a *GREEN* interfészt, akkor egy hozzá tartozó *IP* címet kér a gép. Mivel ez a hálózati kártya átjáróként fog működni két hálózati szegmens (a mi belső hálózatunk az egyik, és az internetet biztosító szolgáltatói szerver és vele kapcsolatban álló *RED* interfészünk a másik) között, ezért konvencionálisan `xxx.xxx.xxx.001` utótagú címet szokás adni neki. Gyakorlatilag rajtunk múlik, hogy az ún. privát címek közül melyiket adományozzuk ennek a kártyának, de érdemes a későbbi fejlesztések miatt következetesen használni őket.
Használható privát címtartományok az *RFC-1918* szabályzat értelmében:

10.0.0.1 – 10.255.255.254 vagy
172.16.0.1 – 172.31.255.254 vagy
192.168.0.1 – 192.168.255.254

Így például ha a a saját belső hálózatunk az ***192.168.1.2***, ***192.168.1.3***, ***192.168.1.4 IP*** című gépekből áll, akkor az *IPCop GREEN* interfészét ***192.168.1.1***-re állíthatjuk, míg például a vezeték nélküli hálózatot használó gépeink ***192.168.0.2*** , ***192.168.0.3*** stb. címűek, akkor az őket összekötő *BLUE*-t ***192.168.0.1***-re (leegyszerűsítve a szétválasztás azt jelenti, hogy jobbról nézve legalább az utolsó előtti számnak eltérőnek kell lennie a különböző interfészek ill. hozzájuk csatlakozó hálózatok esetén).
A rendszer ezután telepíti a betöltéshez szükséges alkalmazást, majd egy információs képernyőn tájékoztat minket arról, hogy böngészőből hogyan lehet elérni az adminisztrációs felületet. Billentyűzetkiosztást (***hu***, vagy ***hu101***), majd időzónát (***Europe/Budapest***) választhatunk a következőkben, végül az *IPCopot* futtató számítógépet keresztelhetjük más névre (alapértelmezettként ***ipcop*** lesz), és amennyiben szükségét érezzük, megváltoztathatjuk a tartománynevet (***localdomain***) is.
Ha nincs *ISDN*-kártya a gépben, ill. nem szeretnénk azt használni, válasszuk az *ISDN* tiltva nyomógombot a következő képernyőn, ellenkező esetben a protokollt és a kártya típusát, valamint a telefonszámot is meg kellene adnunk ahhoz, hogy *ISDN* vonalon keresztül érjük el az internetet.
Ismét egy Hálózati beállítás ablak köszönt ránk, melyben első lépésként a ***Hálózati konfiguráció*** típust kell megváltoztatni *GREEN + RED* formátumúra (jelen esetben nem jó nekünk az alapértelmezett *RED* is ***modem/ISDN***, hiszen kábel vagy *DSL* típusú elérést szeretnénk beállítani). Ha ezzel megvagyunk, a ***Driver*** hozzárendelése kártyákhoz menüpontot aktiválva kiválasztjuk a még szabad másik hálózati kártyát, s ő lesz a *RED* interfész.
Következhetnek a ***Cím beállítások***, s azon belül is a *RED* interfész, mivel ő még nem kapott *IP* címet.
A beállítási lehetőségek ebben az ablakban (a szóközbillentyűvel aktiválhatók az egyes lehetőségek, s csak egy választható ki):

- ***Statikus***: a szolgáltató úgynevezett fix *IP* címet biztosít számunkra, azaz az internet felől nézve mindig ugyanazzal a címmel rendelkezünk. Egyszerű a feladat: ezt a kapott címet kell beírni, a ***Hálózati mask (netmask)*** minden bizonnyal jól van kitöltve.
- *DHCP*: ebben az esetben dinamikusan osztja ki egy szerver számunkra a címet, s ennek változtatásáért és karbantartásáért is ő felel, ezért csak a szolgáltató által megadott számítógép nevét kell a *DHCP* gépnév mellé beírni.
- *PPOE*: tipikusan *DSL* típusú kapcsolatoknál találkozunk vele, s mivel a telefonvonal egyértelműen azonosít, nincs is semmi teendőnk (arra azért ügyeljünk, hogy esetleg a szolgáltató igényli bizonyos beállítások, például speciális gépnév vagy tartománynév használatát, s akkor ezeket a korábban írt lépések során figyelembe kell venni).
- *PPTP*: kábelmodem esetén (is) előforduló típus, itt is csak a szolgáltatótól kapott *IP* címet kell beírni.

A ***DNS és Átjáró*** beállítások menüpontban beállíthatjuk a *DNS* szolgáltatásokért felelős szerverek elérhetőségét, ill. az átjárót is, amennyiben nem *DHCP*-vel kapjuk meg ezeket az címeket. Ezek beállításában is hagyatkozzunk az internetszolgáltatótól kapott információra.



1. ábra Grafikus felületen könnyű dolgunk van

Ha a Rendben gomb megnyomásával túljutottunk ezen a szakaszon, csatasorba állíthatjuk az *IPCop DHCP*-szerver szolgáltatását az Engedélyezve kipipálásával. Ennek segítségével ő osztja ki dinamikusan a belső hálózat gépei számára az *IP*-címeket. Mi egyelőre hagyjuk ki ezt a lehetőséget.
Végül a rendszerünkben mindenható *root* rendszergazdának, s a működtetésért felelős adminisztrátori *admin* felhasználónak adhatunk belépési jelszót. Válasszunk kellemetlenül bonyolult, számok és betűk (kerüljük az ékezeteket ha lehet, mert a billentyűzetkiosztások csúnyán megtréfálhatnak bennünket) kombinációjából összeállított, lehetőleg hosszú és nem szokványos megoldásokat keressünk – mint amilyen például a *130r0tvaha13t0rta*.
Ezzel gyakorlatilag készen is vagyunk, az *OK* megnyomása után hátradőlhetünk: szinte tovább tart végigolvasni ezt a leírást, mint végigcsinálni a folyamatot. Azért teljesen ne bízzuk el magunkat, ha ugyanis az újraindításkor a *CD*-olvasóban felejtettük a telepítőlemezt, „negyedórával korábban mintha ezt már láttam volna” érzés keríthet hatalmába, ezért egyszerűbb a *BIOS*-ban átállítani a betöltési sorrendet a merevlemezzel kezdődőre.

Áll a fal, de hogyan tovább?

A klienseken be kell állítani az *IPCop GREEN* interfészhez tartozó *IP* címét átjárónak, később a szolgáltatások elindítása és a beállítások finomhangolása után érdemes *DNS*-szervernek is megadni, s elméletileg tűzfalunk indulása után már elérhető az internet.
Az induló rendszer négyféle kernelváltozat közül enged választani, nekünk most jó lesz az alapértelmezett első. Betöltés után konzolos ablak fogad, látszólag nem kényezteti el a felhasználót. Hol itt akkor az egyszerűség, könnyű használat, felhasználóbarátság?
Az *IPCop* elsődlegesen böngészőből adminisztrálható, átlátható és jól beállítható felülete miatt népszerű azok körében is, akik nem szívesen irkálnak át mindenféle konfigurációs fájlokat kézzel. Ennek elérését legegyszerűbben így tehetjük meg:
Indítunk egy internetböngésző programot, és a címsorba beírjuk

```
https://192.168.1.1:445/
```

A *https* a biztonságos, titkosított *SSL* kapcsolatot jelzi, s fontos a 445-ös portszám beírása is a kettőspont után. Az idézett *IP* cím (*192.168.1.1*) helyett jó esetben a telepítés során megadott gépnév (például *ipcop*) is használható, de mindenképp azt a címet vagy nevet kell beírni, ami a *GREEN* interfészhez tartozik, tehát a belső hálózat gépeivel azonos hálózati szegmensen található.
A *Connect*-re kattintva beírhatjuk az *admin* felhasználónevet, és a hozzá tartozó jelszót, s máris beléptünk a tűzfalszerverbe.

A nyelvet magyarra is állíthatjuk a *System* menüpont *GUI Settings* pontját választva, de ha a későbbiekben szeretnénk egyéb funkciókat is ellátó csomagokat telepíteni, akkor fontoljuk meg – legalább ideiglenesen – az eredeti angol nyelvű szövegek visszaállítását, mert jócskán megkönnyíti dolgunkat a dokumentációk olvasása során.
Amennyiben valamiért szükségünk lenne átírni az eredeti konfigurációt, például megváltoztatni a hálózati beállításokat, akkor az *IPCop* konzolos felületére belépve a *setup* parancs beírásával a telepítéskor feljövő ablakok szűkebb változatának segítségével megtehetjük azt.
Ha tartunk attól, hogy nagy étvágyú és rosszindulatú egyének megeszik pingvinünket vacsorára, töltsük le az aktuális programfrissítéseket. Ezt a *Rendszer* menü *Frissítések* pontja alatt mazsolázva tehetjük meg, ahol tételesen felsorolva látjuk a megjelent és említésre méltó változásokat, s jobboldali oszlop Információ feliratú linkjére kattintva le is tölthetjük azokat. A lementett fájlt ugyanezen lapon alul a *Tallózást* választva tölthetjük fel, majd az *IPCop* megvizsgálja szavahihetőségét, s telepíti azt.
Munkánkat grafikonok is segítik, valamint szűrhető tűzfal- és egyéb naplózás is a rendelkezésünkre áll. Barátkozzunk a rendszerrel, s olvassunk a témába vágó elméleti és gyakorlati aspektusokkal egyaránt foglalkozó szakirodalmat, hiszen egy igazán jó védelmi rendszernek és üzemeltetőjének illik naprakésznek lennie.
Ráadásul a java még csak most következik, hiszen a feltelepített konfigurációt hamar testre akarjuk majd szabni, s elvárásainkkal összhangban szorgalmunknak is növekednie kell.

Tóth Virgil Zoltán
(m_v@c2.hu)
Szoftverfejlesztő informatikus és rendszergazda, kedvence a Debian disztribúció. Szabadidejét legszívesebben felesége és szépirodalmi regények társaságában tölti. Lenyűgözőnek tartja a Linux rugalmasságát, és a vele dolgozók aktivitását.

Az Amilo7620 notebook nem szereti a Pingvint?

Pár hónappal ezelőtt vásároltam egy Amilo 7620 típusú notebookot ami igen jó vételnek bizonyult, de a Linux telepítéssel igencsak megszenvedtem. Azoknak szeretnék egy kis segítséget nyújtani akik hasonló cipőben járnak.

Szükségem volt, egy notebookra amin bárhová magammal cipelhetem a munkáimat ezért elkezdtem nézelődni a különböző gyártók honlapján, hogy megtaláljam a számomra megfelelő gépet. Vallási okokból kifolyólag a legfontosabb szempont az volt, hogy ***AMD*** processzor és ***ATI*** videokártya legyen a gépben, valamint legalább 512 Mbyte RAM. Így bukkantam rá az ***Amilo 7620*** -ra ami az alábbiakat tartalmazza:

- Processzor: ***AMD Athlon XP-M 2600***
- 512 Mbyte ***RAM***
- 60 Gbyte ***HDD***
- ***ATI Mobility Radeon 9000***
- 10 Mbps hálózati csatoló: ***Via Rhine II***
- Hangkártya: ***Via AC97***
- ***DVD-R, CD RW***
- ***Firewire***
- 5 db ***USB*** csatoló

Ezek a paraméterek számomra nagyon szimpatikusak voltak, de ami a végső lökést adta a megvásárlásához, az ára volt. A hozzá tartozó notebook táskával együtt 165 ezer forintot hagytam a boltban és ez az ár az ***ÁFÁ***-t is tartalmazza. Hasonló teljesítményű hordozható vasak ára 200 ezer forint felett kezdődik (***Sony Vaio*** és társai). Galád módon az eladónak szegeztem a kérdést, mi az átverés benne? Kicsit meghökkent, de aztán elmondta, hogy a gép azért tartozik az alsó árkategóriába, mert nem annyira szép mint mondjuk egy ***Sony*** és a súlya is lényegesen több. Azért ne egy beépített szekrényre gondoljunk...
A kissé túlsúlyos dobozzal a kezemben hazatértem és azonnal neki is ugrottam a csomag – nem túl szabályos – szétcincálásának.
A dobozban megtaláltam a nélkülözhetetlen ***Windows XP Home Edition***-t, a hozzátartozó meghajtóprogramokat tartalmazó lemezeket, egy eredeti ***Nero CD/DVD*** égetőt, a gép töltőjét és a kézikönyveket.
Mivel a cégnél platformfüggetlen alkalmazásokat fejlesztünk ezért a ***Windows*** telepítésével kezdtem, mert nyűgösebb feladat a ***Windowst*** a ***Linux*** után telepíteni mint fordítva.
A ***Windows*** telepítése mintegy 40 percet vett igénybe a legfrissebb meghajtó programok telepítésével együtt. Ezután kezdődtek az igazi megpróbáltatások.
Amióta megismerkedtem a ***Linuxszal***, csak és kizárólag ***Debiant*** használtam, de mivel a programjainknak színes, szagos környezetben kell futniuk ezért most kivételesen a ***SuSE***-t választottam. Természetesen a ***Debiant*** is nagyon szépen testre lehet szabni, de nem nagyon volt kedvem a ***bootsplash*** konfigurálásához,

az utólagos témák töltögetéséhez. A végfelhasználók számára pedig ezek a dolgok a legfontosabbak.
Betettem hát a telepítő ***DVD***-t a meghajtóba és újraindítottam a rendszert. A telepítő program már a telepítési forrás felismerésénél elhalálozott. Átváltottam konzolra, hogy szemügyre vegyem az események alakulását.

```
hdc: lost interrupt
hdc: cdrom_pc_intr: The drive
➥appears confused (ireason =
➥0x01)
```

Ez az üzenet íródott újra és újra a konzolra, amíg újra nem indítottam a gépet. Gondoltam ha a ***SuSE*** összevissza foltozott kernelje nem tetszik az új gépemnek akkor megpróbáljuk egy ***Debian Vanillával***. Letöltés, ***CD*** írás, újraindítás. A telepítési forrás meghatározásánál ismét lefagytunk. Ekkor már kezdtem dühös lenni, úgyhogy az asztali gépen rákerestem a hibaüzenetre. Egy levelező listán találtam egy pár embert akik hasonló problémával küzdenek, illetve találtam egy kernel

fejlesztőtől származó levél idézetet is amely nagy vonalakban arról szólt, hogy a hiba ismert, de alacsony prioritású, mert van hozzá folt illetve ki lehet küszöbölni. Foltot találtam is, de ahhoz legalább egyszer be kellett volna tudnom tölteni a rendszert egy működő kernellel.
A hiba valahol a megszakítás kezelés környékén volt, ezért újra belecsaptam a telepítésbe, de ezúttal minden lehetséges hibaforrást ki akartam küszöbölni ezért az alábbi kernel opciókat használtam:

```
linux acpi=off noapic nolapic
➥vga=771
```

Próbálkozásomat siker koronázta és végre valahára elindult a telepítés. Remek! 30 perc várakozás után már egy komplett ***SuSE***-t tudhattam a gépemen teljes ***GNOME*** és ***KDE*** rendszerrel együtt.
Újraindítás után az *X* konfigurálása következett, ezért az ***ATI*** oldaláról letöltöttem az új ***fglrx*** meghajtót és telepítettem. A grafikus felület újraindítása után sajnos teljesen lefagyott minden. Reset, karbantartási mód.

A */etc/X11/xorg.conf* fájlon az alábbi módosítást eszközöltem:

```
Option "VideoOverlay" "On"
Option "UseInternalAGPGART"
➥"No"
```

Ezután már kifogástalan volt a grafikus felület, volt ***3D*** gyorsítás illetve működni kezdett a ***DRI*** is. Látványosan gyorsabb lett a grafikus felület.
Az utolsó hibára csak hetekkel később derült fény, amikor egy újraírható ***CD***-t próbáltam újraírni, de a ***K3B*** szerint nem volt ***CD*** író a gépben. Hosszas kutakodás az interneten, majd a titokra fény derült. A telepítéskor fellépő kernel probléma volt a tettes ezúttal is, ezért az optikai meghajtót valamiért nem ismeri fel bootoláskor. Ezen könnyen lehet segíteni: az általunk használt bootmanagerbe állítsuk be ezeket az opciókat:

```
hdc=noprobe hdc=cdrom
```

Most már hónapok óta használom a gépet és semmilyen más probléma nem került elő.

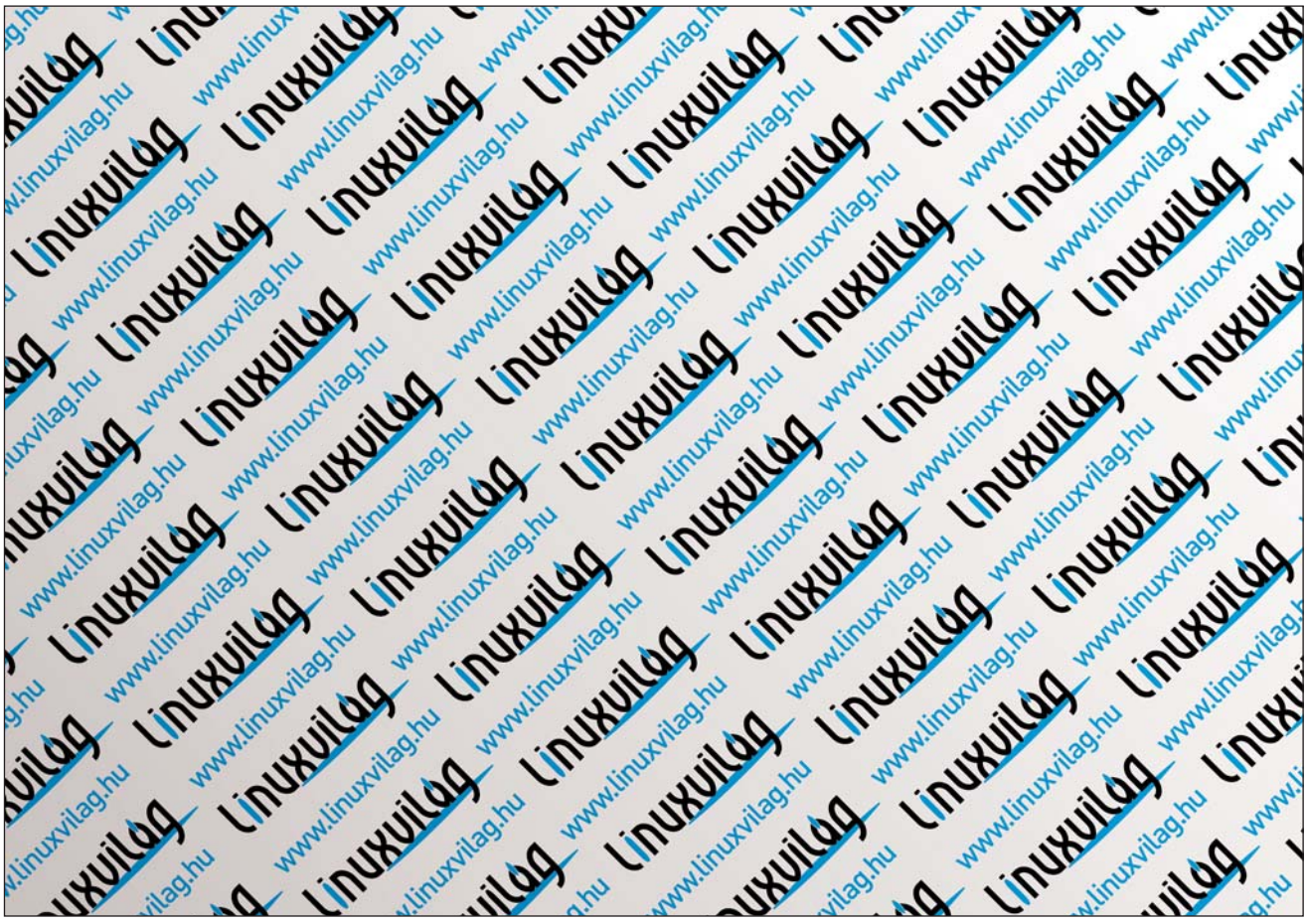
Összegzésként álljon itt a két legfontosabb lépés:

Telepítéshez: `vga=771 acpi=off ➥noapic nolapic`

Bootmanager-be írjuk be telepítés után: `vga=771 hdc=noprobe ➥hdc=cdrom`

A ***Fujitsu-Siemens Amilo 7620*** jelű notebookja kifejezetten jó vétel volt még akkor is ha egy teljes napom ráment míg a ***Linuxot*** sikerült felraknom rá. Sok sikert mindenkinek a telepítéshez!

Tóth Péter
(thotacc@drotnet.hu)
A BMF hallgatója vagyok, mellette egy kis- és középvállalatok informatikai rendszereinek Linuxos átállításával és szoftverfejlesztéssel foglalkozó cég informatikai vezetőjeként tevékenykedem. Kevés szabadidőmet barátnőmmel és barátaimmal töltöm egy-két sör társaságában.



32 bit + 32 bit = 64 bit?

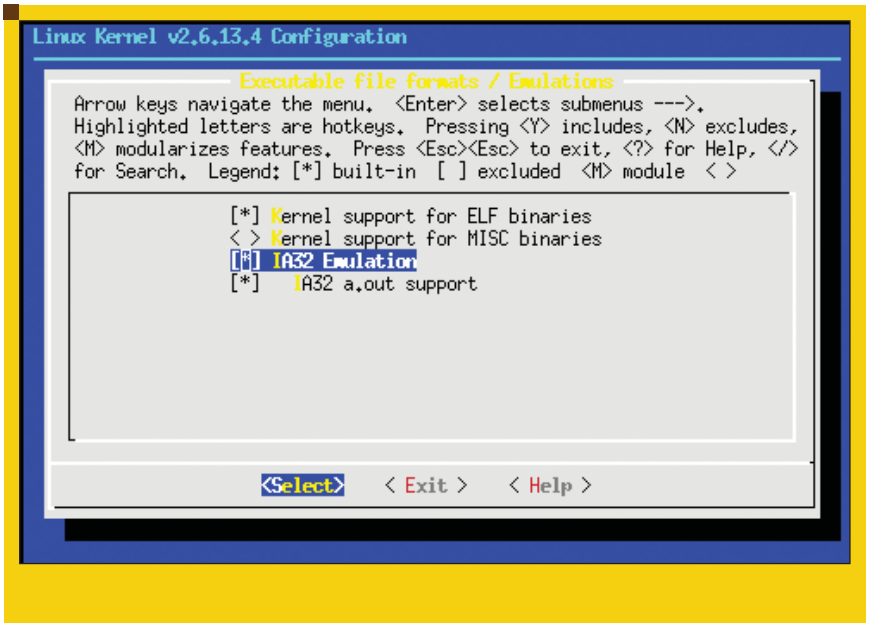
A 64 bites processzorok már jó ideje a piacon vannak, de sokáig csak mint szerverprocesszorokra tekintettek rájuk az emberek, ami nem is csoda, hiszen csillagászati áruk volt. Mostanság azonban már megfizethető közelségbe kerültek, így a hétköznapi ember számára is egyre inkább megfontolandó alternatíva a 64 bites architektúra vásárlása.

Miért jó?

Az már bebizonyosodott, hogy a megahertzek emelése nem mehet túl sokáig, hiszen a frekvenciával arányos a processzor hőtermelése és a fogyasztása. A 64 bit bizonyos tekintetben erre a problémára nyújt megoldás, hiszen a frekvencia maradhat, mégis egy órajel ciklus alatt nem 32, hanem 64 bitet mozgat a processzor, ami bizonyos alkalmazások esetében gyorsulást jelent.
Az kétségtelen, hogy irodai célra nem érdemes 64 bites processzort venni, hiszen egyelőre a 32 bites operációs rendszerek vannak túlsúlyban, arról nem is beszélve, hogy például az *OpenOffice.org-* hoz elég egy mai számítógép is. Aki esetleg mégis 64 bites processzort szeretne vásárolni, jó, ha tudja, hogy a processzor visszafelé kompatibilis, vagyis futnak rajta a 32 bites programok is. Nyilván ebben az esetben le kell mondanunk a nagyobb bitszélesség nyújtotta sebességnövekedésről.
A másik dolog, ami miatt érdemes lehet 64 bites processzort venni, hogy megfelelő alaplappal nem csak 4 gigabyte memóriánk lehet hanem jóval több. (Korábban gyakori volt, hogy az operációs rendszer a négyből csak három gigabyteot látott.). Az *AMD64*-hez például gyártanak olyan alaplapot, ami akár 4 darab 2 gigabyteos *DDR* modult is képes fogadni. Persze egyelőre nem sok olyan otthoni felhasználó van, aki ezt memóriamennyiséget ki tudná használni, a dolognak inkább a tudományos kutatásoknál lehet hasznát venni. Azt már tényleg csak zárójelben merem megjegyezni, hogy a 64 bites architektúrával a megcímezhető memória az eddigi 4 gigabyte helyett körülbelül 16 milliárd gigabytera emelkedik.

EM64T vagy AMD64?

Jelenleg két 64 bites szabvány van a piacon: az *AMD*-től az *AMD64* az *Inteltől* pedig az *EM64T* (*Extended Memory 64-bit Technology*). A kettő nem kompatibilis egymással, amire a 64 bitre optimalizált telepítőkészlet letöltésekor oda kell figyelni. Nem szeretnék hitvitát nyitni arról, hogy az *AMD* vagy az *Intel* a jobb. Mindenki nyilván azt fogja használni, ami neki és a pénztárcájának a legjobban megfelel. Jómagam egy *AMD64* processzor tulajdonosa vagyok, de *Intel* esetén mindenben teljesen hasonlóan kell eljárni.
Talán azt is érdemes megjegyezni, hogy a jelenlegi hardvereink az új architektúrával is használhatóak, amennyiben a gyártó rendelkezésünkre bocsátotta a 64 bites rendszerhez szükséges



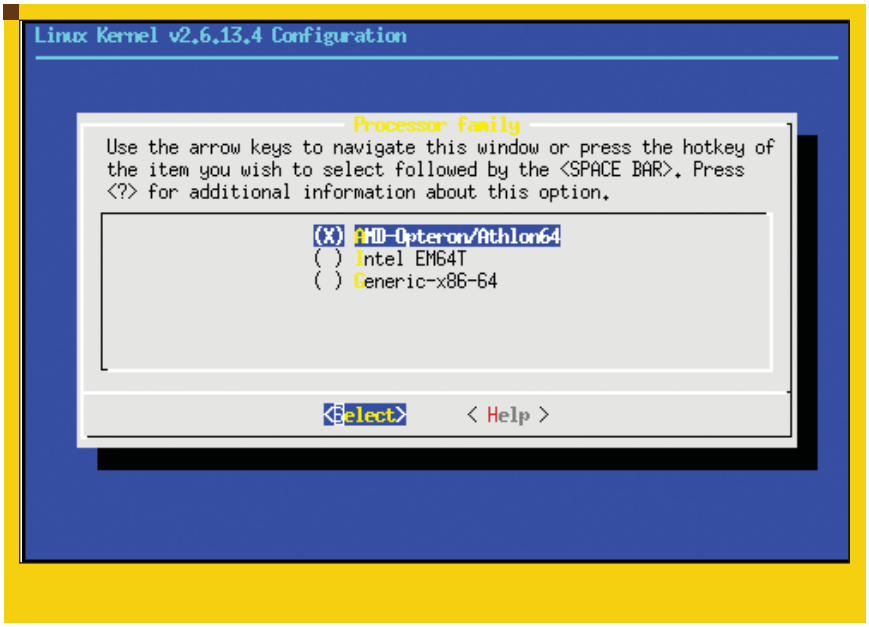
meghajtóprogramokat, vagy azok forrását, amit aztán jó eséllyel le tudunk fordítani 64 bitre.

Operációs rendszer

Szerencsére már számos disztribúció előrukkolt 64 bites támogatással, igaz sok még csak kísérleti stádiumban van (ilyen például a *Debian*). Jómagam két disztribúciót telepítettem fel: a *Debiant* és a *Gentoot*. Ez utóbbi még mindig tesztelés alatt áll. Akinek nem volt még dolga *Gentoo Linuxszal* az jó ha tudja, hogy itt a felhasználó a teljes rendszert lefordíthatja magának a saját gépére optimalizálva. Jómagam a fordítással eléggé le vagyok maradva – lévén a *Gentoo* nálam amolyan mostohagyerek, inkább a *Debiant* használom mindennapos operációs rendszernek.
Érdemes persze körülnézni kedvenc disztribúciónk háza táján, sose lehet tudni, hátha abból is megjelent a 64 bites verzió.

Debian és az AMD64

A *Debian AMD64*-es változata még kísérleti stádiumban van és állítólag csak a következő változatban (*Debian Etch*) lesz benne hivatalosan. Ez persze nem jelenti azt, hogy használhatatlan, hiszen én magam is napi rendszerességgel használom és eddig semmi rendellenességet nem tapasztaltam.



A netes telepítést választottam, így csak egy telepítő CD lapul a fiókomban. A bootoláskor a hálózatot felismerte, így az *i386*-os architektúránál megszokott módon telepítettem a rendszert. A netes telepítéshez és a későbbi frissítésre én az ➲ *ftp.hr.debian.org (horvát)* tükröt javasolnám, ez van ugyanis a legközelebb és ez bizonyult a leggyorsabbnak is.
Azzal a gondolattal azonban meg kell barátkoznunk, hogy 64 bitre még nem érhető el minden program. De erre is mutatok rögtön megoldást.

Kernel a'la 64 bit

Természetesen minden telepítés végén érdemes saját kernelt fordítani. Jelen esetben két dologra kell nagyon odafigyelni: az egyik, hogy milyen processzorra fordítjuk (*Intel* vagy *AMD*), illetve hogy az *Executable file formats / Emulations* menüben jelöljük be az *IA32 Emulációt*. Ez utóbbi jelenőségére később még kitérek. Bár a 2.4-es kernelben is van 64 bites támogatás, mégis valamelyik 2.6-os kernel használatát javaslom.

Néha sok a 64 bit...

Telepítés után rögtön volt egy olyan hiba, amit pár percig egyszerűen nem értettem. Volt egy futtatható állomány, hozzá futtatási jog, a *Bash* parancsértelmező mégis azt mondta, hogy a fájl nem található. A jelenséget az okozta, hogy a *Linux* megpróbálta 64 bitesként értelmezni a 32 bites állományt, de ez így alapból nem megy.
Létre kell hozni egy 32 bites *chroot* környezetet, amelyben aztán minden régi programunkat tudjuk futtatni. Ehhez kell – többek között – a korábban említett 32 bites emuláció bekapcsolása a kernel fordításkor. Lássuk tehát, hogyan kell felállítani egy 32 bites *chroot* környezetet. Legelőször telepítsük fel a `debootstrap` csomagot. Ha ez megvan, gépeljük be az alábbi parancsot:

```
debootstrap –arch i386
➥sarge /emul/ia32-linux
➥http://ftp.hu.debian.org/
➥debian/
```

Ez feltelepíti a */emul/ia32-linux* könyvtárba az alap *i386*-ra való *Debiant*. Ezután lépjünk be ebbe a rendszerbe a

```
chroot /emul/ia32-linux
```

paranccsal, majd telepítsük az összes szükséges csomagot az `apt-get` parancs segítségével. Ilyen például a `libx11-6` csomag.
Ha ezzel megvagyunk, akkor a 64-bites (gazda) *Debian /etc/ld.so.conf* fájljába vegyük fel az alábbi 32 bites könyvtárakat tartalmazó mappákat:

```
/emul/ia32-linux/lib
/emul/ia32-linux/usr/lib
/emul/ia32-linux/usr/X11R6/lib
/emul/ia32-linux/usr/local/lib
```

Ha ezzel megvagyunk, futtassuk le az `ldconfig` parancsot, amivel frissítjük a könyvtárakat tartalmazó gyorstárat. Ezzel persze még nem vagyunk készen, hiszen a *chroot* környezetben nincs például *home* könyvtárunk. Hogy ez is meglegyen, a 64 bites rendszerünk */etc/fstab* fájljába írjuk be a következő sorokat:

```
/home   /emul/ia32-linux/home
➥none   bind 0 0
/tmp   /emul/ia32-linux/tmp
➥none   bind 0 0
proc   /emul/ia32-linux/proc
➥proc   defaults   0 0
```

Hozzuk létre a megfelelő helyeken a könyvtárakat, majd mountoljuk fel.
Nyilván érdemes létrehozni azokata felhasználókat a *chroot* környezetben is, akik használni fogják a rendszert. Ezt a közismert `useradd` paranccsal tehetjük meg. Arra oda kell figyelni, hogy a *chroot* környezetben és a normál környezetben minden felhasználónak ugyanaz legyen a *felhasználói* (*UID*) és *csoportazonosító* (*GID*) száma, hiszen közös lesz a kétféle *home* könyvtár. Végül, ha ezzel is megvagyunk, a gazda operációs rendszerre telepítsük fel a `dchroot` csomagot, így nagyon könnyen lehet váltogatni az 32/64 bites környezetek között.

Chroot – gép a gépben

A *chroot* segítségével oldható meg, hogy egy gépen, sőt egy partíción belül több operációs rendszer „lakjon”. Ezek *virtuális operációs rendszerek* egy gazda operációs rendszeren belül. Ez azonban nem keverendő össze sem a *Bochs*, sem a *Xen*, sem a *VMware* programokkal. Ott ugyanis megoldható az operációs rendszerek keverése, itt nem. Ez a megoldás akkor jó, ha ki szeretnénk próbálni mondjuk a *Debian* legújabb verzióját a régi letörlése vagy a merevlemez particionálása nélkül. A *chroot* akkor is jól jön, ha egy gépre több felhasználónak kell különböző *Linuxot* telepítenünk, például azért, mert a munkájához mindenkinek szüksége van rendszergazdai jogokra is. Ha ilyenkor nem akarjuk, hogy egymás dolgaiba belelássanak, a *chroot* környezet kialakítása a megoldás. Persze ilyenkor is van *főrendszer,* amit karban kell tartani valakinek, de az egyes *alrendszereket* az „alrendszergazdai jogokkal” bíró felhasználók tarthatják karban.

Egyszerűen kiadjuk az alábbi parancsot:

```
dchroot -c ia32 -d bash
```

Ezzel máris egy 32 bites *Bash*-ban találjuk magunkat. Természetesen ugyanezzel módszerrel bármilyen más, a *chrootban* létező programot is elindíthatunk. Ahhoz azonban, hogy ez lefusson, a */etc/dchroot.conf* fájlnak tartalmaznia kell az alábbi sort:

```
ia32 /emul/ia32-linux
```

A `dchroot`-ot nem csak 64 bites X alatt lehet futtatni. Szöveges konzolról, majd – ha a bash indítottuk – elindíthatunk egy teljesen 32 bites X-et is a

```
startx -- :1
```

paranccsal. Ezt az X-et a megszokott *Ctrl+Alt+F7* helyett a *Ctrl+Alt+F8* billentyűkombinációval érhetjük el. Néhány ok, amiért a 32-bites *chroot* hasznos:

- Az *OpenOffice.org* lefordítása (64 bitre, de akár bármilyen platformra) rendkívül időigényes feladat, minek pazarolni az időt, ha már létezik 32 bitre.
- A cikk írásának idején a *Macromedia Flashplayer* nem rendelkezett 64-bites támogatással. A *chroot* segítségével azonban nem kell mellőznünk a weboldalak *Flash* animációit. Vannak sajnos olyan oldalak, amiket *Flash*-támogatás nélkül nem is tudunk megnézni.
- Megy alatta a *Wine* és az *Acrobat Reader* is.

Nyilván az említett *chroot* környezetet létre lehet hozni más disztribúcióban is a fentiekhez hasonló módon.

Medve Zoltán
(e-medve@e-medve.hu)

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

64 bit
➲ http://en.wikipedia.org/wiki/64-bit

Debian telepítő médiák
➲ http://cdimage.debian.org/cdimage/

Debian AMD64 médiák
➲ http://cdimage.debian.org/cdimage/unofficial/sarge-amd64/

AMD64-hez csomagok
➲ ftp://ftp.hr.debian.org

Egy számítógépterem „évköznapjai”

Egy számítógéptermet – főleg iskolai környezetben – nem elég csak talpra állítani. Üzemeltetés közben akadnak meglepetések, és folyamatosan fejlődnek a háttérben álló programok is. Ezek tanulságaiból olvasható itt egy kisebb csokorra való.

A QEMU-ról még egyszer

2005 szeptemberében egy *QEMU*-ról szóló cikkben (***Linuxvilág #56/39. oldal***) szó esett arról, hogy bináris formában meglevő régi programjainkat (például *Maple*) el lehet indítani egy emulátorban, például a *QEMU*-ban. Ezt az írást szeretném kiegészíteni: akkoriban a *QEMU* gyorsítását végző *kqemu* kernelmodult (a */dev*-beli bejegyzés nehézkes létrehozása miatt) elég körülményes volt betölteni – erről beszámol a cikk is. Azóta az udev rendszernek (és a kqemu-0.7.2 megjelenésének) köszönhetően ez elegánsabbá vált. Elegendő egy bejegyzés a */etc/udev/permissions.d/50-udev.permissions* fájlba:

```
kqemu:root:root:0666
```

Ekkor a

```
modprobe kqemu
```

már minden gond nélkül betölti a gyorsítómodult.

Mennyire törhetetlen az ext3?

A 2005 áprilisában jelent meg a *Linux-sokszorosítás iskolai gépekre* (***Linuxvilág #51/47. oldal***) című írás. Azóta több olyan cikk jelent meg az újságban, amiről örömmel láttam, hogy nagyszerűen kiegészíti ezt (és viszont). Az októberi számban (***Linuxvilág #51/27. oldal***) részletesen olvashattunk a *System Rescue CD*-ről. Magam is ezt használom partíciók kezelésére (pontosabban ezen a *partimage* programot). Fontos tanulsága volt az elmúlt heteknek, hogy ennek a *LIVE CD*-nek (vagy a winchesterre telepített változatnak) a leállítása előtt nem célszerű az

```
umount -a
```

parancsot kiadni (pedig ez igen csábító lehetőség a szkriptből való partíciómásolás, felcsatolás és a rajta való tényked ás után). Ha a frissen feltöltött partíciót felcsatoltam, frissen odamásoltam egy *menu.lst* fájlt, végül az `umount -a` paranccsal le akartam csatolni „mindent”, akkor újraindításkor gond volt a *GRUB*-bal (csak annyit írt ki, hogy `GRUB`). S ez nemcsak néhány esetben volt így, hanem 30 gépből mind a 30 gépen így történt. Az `umount -a` valójában nem tudott minden fájlrendszert lecsatolni – hibával leállt, és nem nyúlt hozzá az általam felcsatolt merevlemez-partícióhoz sem, ilyen módon pedig sérült a fájlrendszer. (Ez azért érdekes tanulság, mert korábban az a kép élt bennem, hogy az `umount` utasítás nélkül is szinte mindent túlél az *ext3* fájlrendszer.) Kiderült, hogy ilyenkor explicite érdemes megadni, ha valamelyik partíciót le szeretnénk csatolni.

A pillanat, ami eljött

A 2005 szeptemberi számban jelent meg (***Linuxvilág #56/51. oldal***) *Medve Zoltán* írása *Nagyban kicsit más – Linux az oktatási szférában* címmel. Az 53. oldalon kilátásba helyezi, hogy *„már ígérik, hogy a GDM-nek is lesz olyan szolgáltatása”*, hogy a grafikus felületre való bejelentkezés előtt

elindulhasson egy (lebutított) böngésző, például az üzemeltetéssel kapcsolatos tudnivalókról.
Nos, ez a pillanat most eljött: ***UHU 1.2*** alatt sikerült ezt megvalósítani ***GDM*** alól.
Mindössze a ***/etc/X11/xdm/Xsetup*** konfigurációs fájl végére kellett odaírni azt a sort, ami elindítja a böngészőt egy induló ***HTML*** fájllal. Apró kényelmetlenség ugyan, hogy ha hozzányúlunk a böngészőhöz, akkor nehéz visszakapni a bejelentkezési kurzort. Egyelőre ezt úgy tudom megoldani, hogy kérek egy menüpontot (pl. ***Műveletek*** – lent) , majd a ***Mégsem***-re kattintok (úgy, hogy az egér ne legyen a böngésző fölött), és újra ott villog a vékony kurzor a helyén.

Esélyegyenlőséget!

Belső hálózatunk webszerverén két kisebb ***PHP*** programmal próbáltam „esélyegyenlőséget teremteni”, már ami az egyes felhasználók gép mellett töltött idejének mennyiségét illeti.
Az egyik egy térképszerű táblázattal mutatja meg, hogy melyik gépre ki jelentkezett be és mikor. Ehhez csak a ***/etc/passwd*** (és esetleg az csoportinformációt hordozó ***/etc/group***) fájlt kell olvasnia a programnak, valamint a szervergépen indított `w` parancs kimenetét kell megfelelő módon kell feldolgozni.
Egy olyan adatfájllal, amely soronként tartalmazza a bejelentkezési nevekhez tartozó fényképek fájlneveit, arcképe is megjeleníthető. A bejelentkezés óta eltelt idő alapján átszínezhetjük az egyes felhasználók arcképe mögötti táblázatcellát, ezzel segítve a gépterem felügyelőjét, hogy a túl régóta jelenlevőket figyelmeztethesse, ha éppen sokan várakoznak.

Tükröm, tükröm, mondd meg nékem...

Nálunk ***Samba*** alapú hitelesítéssel lépnek be a diákok mind ***Linux***, mind ***Windows*** alól a hálózatra. Bár – némiképp különös módon – ez a program meglehetős népszerűségnek örvend, előfordul, hogy egy-egy diák elégedetlen a (többnyire diákigazolványból származó) portréjával.
A másik programocska az arcképcsere lehetőségét szolgálja. Ilyenkor természetesen nem automatikusan cserélődik le a kép, nehogy felismerhetetlen portré cserélje le a felismerhetőt, vagy ad abszurdum más login nevével bejelentkezve megtréfálják az illetőt.
Az érdeklődők a ***Linuxvilág*** webhelyéről szabadon letölthetik a két említett program forráskódját.

Démonok, klónok, meg minden

Gyakran kerülök abba a helyzetbe, hogy – bármily gondosan gondoltam is át, hogy milyen programokra lesz szükség az év folyamán – kisül, hogy egy-egy informatikai szakkörön vagy tanórán bizony még szükség lenne egyebekre is, amik nincsenek telepítve.
Ilyenkor a „klasszikus” megoldás az, hogy egy gépet helyrehozunk, majd klónozással a többi gépre is felvisszük a változásokat. Ez persze meglehetősen időigényes, és erre általában nincs napközben lehetőség. Főleg akkor nem, amikor tele a gépterem, és szeretnénk haladni a témával, csak épp valami hiányzó csomag miatt nem fordul le egy program.
Az ilyen esetek miatt eszeltem ki a következő apró cselt.
A rendszerbetöltéskor elindítok a háttérben egy héjprogramot (kvázi „démont”), ami csak annyit tesz, hogy tíz másodpercenként ránéz egy (***Sambával*** megosztott) hálózati

1. lista A *menu.lst*

```
default saved          # Fontos!!!

title the old kernel # vagy: az általában futtatott oprendszer
root (hd0,0)
kernel /old_kernel
savedefault

title the new kernel # vagy: a ritkán futtatott oprendszer
root (hd0,0)
kernel /new_kernel
savedefault 0          # Fontos!!!
```

könyvtárban levő fájlra (nevezzük ezt „retesz”-nek). Ha az ott van, akkor lefuttat egy másik (szintén hálózati könyvtárban levő) héjprogramot. Persze mindkét hálózati fájlt csak az adminisztrátor tudja írni vagy törölni (vagy létrehozni). Nyilván először létrehozom a futtatandót, és ha kész, csak akkor érintem meg (`touch`) a „reteszt”.
A lefuttatott program általában annyit tud, hogy ciklikusan megvizsgálja, megvan-e a */tmp* könyvtárban a „futottam-már” jelzőfájl. Ha nincs, akkor a `-y` opcióval (hogy ne kelljen a felmerülő telepítési kérdésekre manuálisan válaszolni) és a semmibe irányítva a kimenetet és a hibákat, letölt egy-két csomagot (például `apt-get -y install libidn-dev qt-designer>/dev/null 2>&1`), majd létrehozza a „futottam-már” jelzőfájlt a */tmp*-ben.
Amikor 10 másodperc múlva újra megvizsgálja a „futottam-már” fájlt, már ott találja, és nem kísérli meg újra a csomagtelepítést.
Nyilván hasonlóképp bármi más parancs is lefuttatható. Kiadható például egy `halt` parancs, ami megspórolja a gép kikapcsolásához tartozó egyéb fárasztó teendőket és azokat a buzdításokat, amik a diákok késlekedését próbálják lefaragni. Főleg kisebb diákjaim igen szeretik azt a megoldást (tanóra végén), hogy az arra kiszemelt ember létrehoz a(z írható) hálózati könyvtárában egy „kikapcs” fájlt. Persze előzőleg én már elindítottam egy ciklust, ami néhány másodpercenként ránéz erre a fájlra, és ha megvan, akkor adja ki a – mindenkire ható – `halt` parancsot.

Ami elromolhat, az el is romlik

Nem is gondolná az ember, hányféle módon lehet elrontani ezt az egyszerű folyamatot. Például fájl helyett könyvtárat hoznak létre, vagy nagy betűvel hozzák létre: „Kikapcs”, vagy lecsatolják a könyvtárat stb. A tanári gépen célszerű előtte kilőni ezt a „démont”, nehogy az is áldozatul essen a kikapcsolási hadjáratnak. Érdemes arra is vigyázni, nehogy olyan állapotban maradjanak a hálózati fájlok, hogy a legközelebbi óra elején a betöltés után egyből kezdetét vegye az össznépi kikapcsolás (mint ahogy már jártam).
Ha egy kicsit több időnk van (és esetleg egy-két program telepítésénél vagy fájlok másolásánál komolyabb változtatást eszközöltünk), akkor érdemes újraklónozni a gépeket. ***Koblinger Egmont*** megoldása igen elegáns: egy gépet helyrehozunk, majd ***MD5***-vizsgálattal megállapítjuk, mennyiben tér el a klónozandó partíció (amit már használtak a felhasználók) a vágyottól – ezután csak a különbséget kell gépenként feltöltögetni.
Ennél kicsit lassabb egy másik módszer, ami arra alapoz, hogy azt a merevlemezt, amelyikről másolunk, mint egyik szűk keresztmetszetet kiküszöböljük: a partíciók másolásának elkezdése előtt egy héjprogram ciklusa másodpercenként figyeli egy (hálózaton található) fájl meglétét. Mikor minden gép kész a másolásra, csak akkor hozom létre ezt a fájlt, és ekkor kezdődik a ***partimage*** futtatása. Ekkor nyilván túlnyomó részt a gyorsítótárból tudja adagolni az információt a szervergép. Sokkal gyorsabban végzek így egy gépterem telepítésével, mintha sorban indítanám (10-20 másodperc különbséggel) az egyes gépeken a klónozást. Persze a csomagok ütközése sokat lassít, de azért egy 5 GB-os partíció 30 gépre való másolása 15 perc alatt lezajlik így.

A GRUB képességeiről

Szép lehetőség a ***GRUB*** azon opciója, hogy megadható, hogy legközelebb melyik menüpontnak megfelelő operációs rendszer induljon. Így nézhet ki a ***menu.lst*** fájl (*1. lista*).
Amikor olyan operációs rendszert (vagy új kernelt) szeretnénk indítani, amit csak egyszer kell futtatni (például a klónozáshoz szükséges ***System Rescue CD*** merevlemezen levő változatát), akkor előzőleg kiadjuk a

```
grub-set-default 1
```

parancsot.
Ekkor a másik opció töltődik be, de betöltéskor már át is állítja a legközelebb betöltendő menüpontot a másikra. Lehetőség van a `fallback` opció használatára is, ha több bizonytalan betöltendő menüpontunk is van...
Lehet távoli fájlrendszeren levő kernelt is betölteni az új ***GRUB***-bal (sőt, a ***menu.lst***-t is el lehet itt helyezni).
A ➲ ***http://www.gnu.org/software/grub/manual/html_node/ helyen, vagy a GRUB felhasználók közössége által szerkesztett ➲ http://www.autistici.org/grub/moin.cgi/FrontPage*** wiki oldalon hasznos részleteket találhatunk ezzel kapcsolatban. Fellapozhatjuk persze a ***Linuxvilág 43.*** és ***45.*** számát is, melyben ***Komáromi Zoltán*** írt a(z akkori) ***GRUB***-ról, s akitől ezeket a hivatkozásokat és ötleteket is kaptam.
Sok türelmet és sikert kívánok minden linuxos gépterem karbantartójának!

Szabó Zoltán
(szz@freemail.hu)
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.

A clapf spamszűrő

Ebben a számban egy saját fejlesztésű, zlib/png licenc (http://www.opensource.org/licenses/zlib-license.php) alatt közreadott, nyílt forrású spamszűrőt mutatok be, amely gyors, könnyen és egyszerűen használható.

A *clapf* (http://freshmeat/projects/clapf/) kezdetben egyszerű antivírus modulként indult, *Postfixhez* kerestem olyan programot, amellyel ki lehet szűrni a vírusos leveleket, és úgy döntöttem, kreativitásomat egy új program elkészítésével fogom kiélni. Később jött az ötlet, hogy legyen képes felismerni és megjelölni a spam-et is. A *clapf* egy olyan démon, amely az nyílt forrású *clamav* (➲ *http://www.clamav.net/*) antivírus könyvtárat használja, és *SMTP* protokoll segítségével kommunikál a *Postfix*-szel (vagy bármely *MTA*-val, amely képes ilyen módon együttműködni vele).

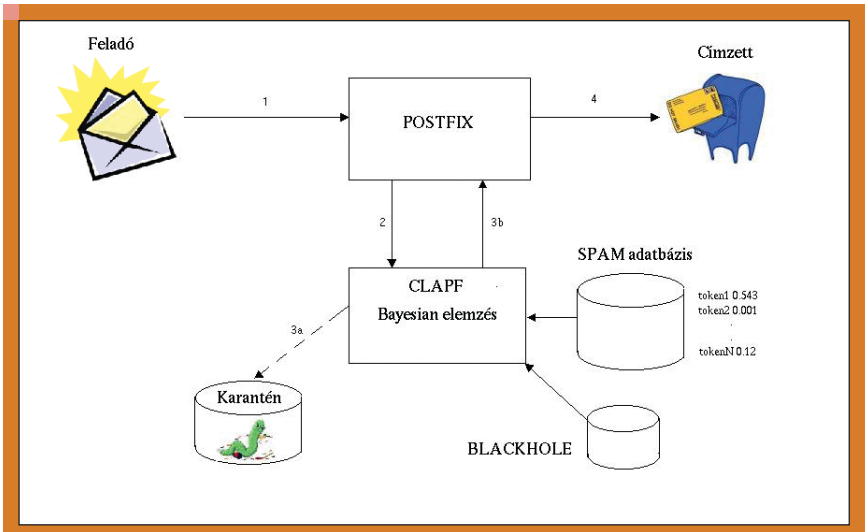
A telepítés

Az alábbi példában a *clapf* a helyi gép (localhost) *10025/tcp* portján várja a leveleket, és a *10026/tcp* portra adja vissza a *Postfix*-nek. Mielőtt a *clapf* telepítéséhez látnánk, installáljuk fel a *clamav*, *tinycdb* (➲ *http://www.corpit.ru/mjt/tinycdb.html*) csomagokat és a *CDB_File* nevű *Perl* modult (➲ *http://www.cpan.org/*).
Első lépésben telepítsük a *clapf*-ot! Hozzunk létre egy dedikált felhasználót, aki a spamszűrőt futtatni fogja:

```
groupadd av
useradd -g av -s /bin/sh -d
/opt/av av
usermod -L av
```

Töltsük le a program legfrissebb változatát, majd csomagoljuk ki:

```
tar zxvf clapf-0.3.20.tar.gz;
cd clapf-0.3.20
```



1. ábra A nagy kép

Igény esetén módosítsuk a *config.h*-ban szereplő változókat, majd fordítsuk le a *clapf*-ot:

```
./configure
make
su -c 'make install'
```

Ha csak spamszűrésre van szükségünk, akkor használjuk a

```
./configure --disable-clamav
```

parancsot.

Hozzuk létre a működéséhez szükséges könyvtárakat:

```
mkdir -p /opt/av/quarantine
chown -R av:av /opt/av
chmod 700 /opt/av
```

A 0.3.19 verziótól kezdve a *clapf* változóinak nagy részét konfigurációs fájlban tárolja. Kiindulási alapként használható a */usr/local/etc/clapf.conf* fájl. Módosítsuk ezt igény szerint, amit megjegyzések bősége segít.
Készítsük el a *Bayesian* elemzéshez szükséges adatbázist. Ehhez szükségünk lesz a *HAM* és *SPAM* nevű *mbox* formátumú (ahol a *Maildir* formátummal ellentétben egyetlen fájlban vannak a levelek) fájlokra, amelyekben csak ham (értékes levél, egyszóval nem spam), ill. csak spam (kéretlen levél) van. Ha ez megvan, akkor az alábbi parancsokkal készítsük el a spam adatbázist (ez valójában egy *CDB* fájl), majd másoljuk azt be a */opt/av* könyvtárba:

```
sh util/kcdb.sh HAM SPAM
mv tokens*.cdb /opt/av
chmod 644 /opt/av/tokens*.cdb
```

Végre, minden készen áll, hogy a spam- és vírusszűrőnk elinduljon (Figyelem! A *clapf*-ot ne futtassuk rendszergazda jogosultságokkal):

```
su - av -c "/usr/local/bin/
➥clapf -c /usr/local/etc/
➥clapf.conf &"
```

A naplófájlban az alábbihoz hasonló üzeneteket kell találnunk:

```
Oct 12 13:26:43 thorium
➥clapf[15851]: reloaded
config: /usr/local/etc/
➥clapf.conf
Oct 12 13:26:43 thorium
➥clapf[15851]: using /opt/av/
➥tokens.cdb as spamicity file
Oct 12 13:26:43 thorium
➥clapf[15851]: clapf 0.3.xx
➥starting
Oct 12 13:26:45 thorium
➥clapf[15851]: reloaded with
➥40544 viruses
```

Ha a *clapf* rendben elindult, módosítsuk a *Postfix* konfigurációs állományait, hogy a leveleket beérkezés után adja át a *clapf*-nak. Ehhez először a */etc/postfix/main.cf* fájlhoz adjuk hozzá az alábbi sort:

```
content_filter =
➥smtp:[127.0.0.1]:10025
```

Majd a */etc/postfix/master.cf* fájlhoz pedig a következőt:

```
127.0.0.1:10026 inet n - n - 10
➥smtpd -o content_filter=
-o receive_override_
➥options=no_address_mappings
```

A *clapf* alapesetben maximum 16 címzettet kezel le (`MAX_RCPT_TO`), amit célszerű a *main.cf*-ben is beállítani (`smtpd_recipient_limit = 16`). Ajánlott limitálni a levelek hosszát is (`message_size_limit`).

Végül indítsuk újra az *MTA*-t az alábbi paranccsal:

```
postfix reload
```

Néhány dolgot nem említettem, hogy például a vírusszignatúrákat rendszeresen (egy konferencián azt javasolták, hogy óránként) frissíteni szükséges. Valahogyan gondoskodni érdemes arról, hogy a *clapf* esetleges leállása esetén újra elinduljon. A *util/* könyvtárban néhány héjprogram (shellszkript) segít ezekben.

Az üzenet útja a felhasználóig

Nézzük meg, hogy egy bejövő üzenet milyen utat jár végig, mire a felhasználó postaládájába kerül. A levél első lépésben a *Postfix*-hez kerül, amely úgy van konfigurálva, hogy azt először a szűrőhöz irányítja. A *clapf* először vírusellenőrzést végez. Ha egy levélben vírust talál, akkor azt `"550 Access denied"` üzenettel utasítja el, amit a *Postfix* közöl a feladó *SMTP* szerverével (az üzenetben a vírus neve is szerepel). Ebben az esetben véget ért a levél útja. Opcionálisan – ha a konfigurációs fájlban a `use_quarantine=1` szerepel – a vírusos levelet karanténba teszi, ahol a rendszergazda később megnézheti, és dönthet felőle. Ha a levélben nincs ismert vírus, akkor jöhet a spam vizsgálat.

A *clapf* először a *Bayesian* formulát alkalmazza a levél tárgyára (`Subject`) és a levél törzsére. *Paul Graham* szerint (➲ *http://www.paulgraham.com/better.html*) jó, ha a teljes fejléc is szerepel a *Bayesian* elemzésben, mert fontos információk lehetnek annak egyéb részében is. Én azonban úgy gondolkodtam, hogy ott leginkább nyílt proxy-kat, fertőzött zombikat találunk, továbbá az időbélyegeknek, queue azonosítóknak sem látom igazán hasznát ebből a szempontból. Egyelőre ezek nélkül is megfelelő eredménnyel dolgozik a *clapf Bayesian* elemzése. Miután a *clapf* megvizsgálta a levelet, egy extra mezőt – `X-Clapf-spamicity` – tesz a fejlécbe, amelyből kiderül, hogy az mekkora valószínűséggel spam. Ha az érték a nullához van közelebb, akkor ham, ha inkább az egyhez, akkor spam, a 0,5 körüli eredmény pedig bizonytalanságot jelent. Akkor működik jól a szűrő, ha a spam valószínűség értékek (spamicity) a 0,001 ill. 0,999 körül képződnek. Az a levél, amelynek fejlécében például az

```
X-Clapf-spamicity: 0.0123
```

sor található, az nagy valószínűséggel ham. Ha a *clapf* úgy találta, hogy spam-mel került szembe, azaz a valószínűség értéke 0,94-nél (`spam_overall_limit`) nagyobb, akkor beilleszt még egy fejlécet a levélbe a kliens oldali felismerés (például *procmail*, *maildrop*) megkönnyítésére:

```
X-Clapf-spamicity: Yes
```

A *clapf* naplózza minden feldolgozott levél azonosítóját (ez valójában egy *MD5* hexa hash formátumú fájlnév, például `4b91c518b39ddaa9c8cf609d1db467`), spam valószínűség értékét, a levél méretét bájtban és a feldolgozáshoz szükséges időt mikroszekundumban.

A *clapf* nem rendelkezik sor (queue) kezeléssel. Az elején terveztem ezt a funkciót, de a probléma meggyőzött arról, hogy fölösleges újra feltalálni a kereket: a *Postfix* mind a *clapf* előtt, mind pedig utána megoldja ezt a problémát.

Szintén egy konferencián hallottam olyan antispam termékről, amely – ha nagyon biztos a dolgában – eldobja a levelet. Ennek megvan az az előnye, hogy a felhasználó postaládájába be sem jut a *spam* – ez különösen

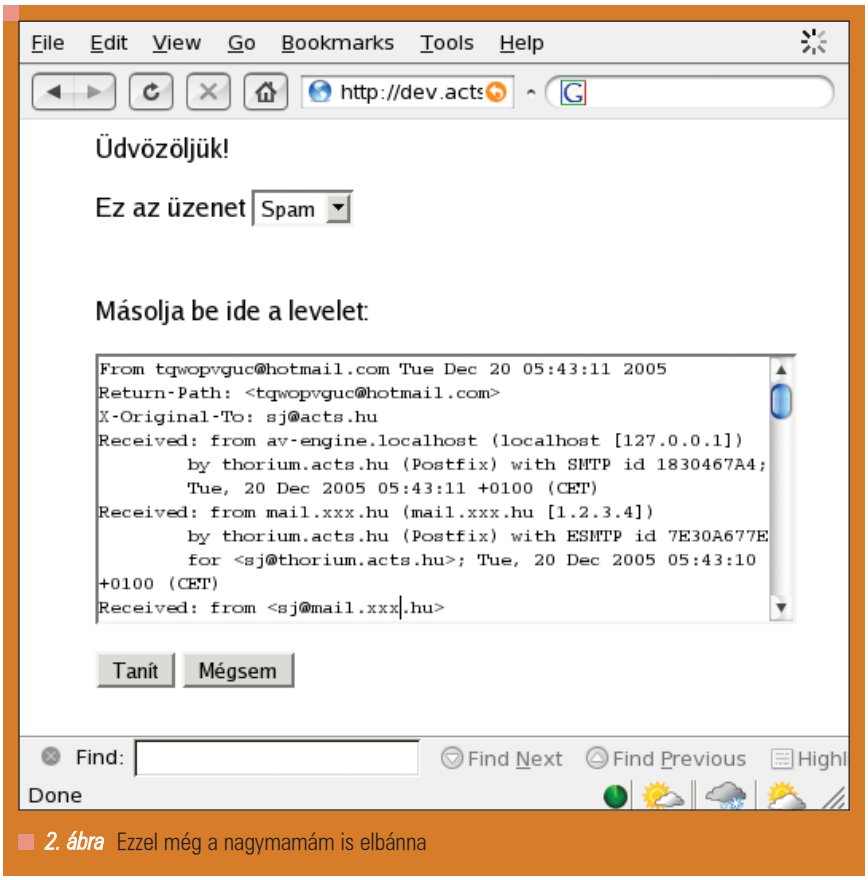
akkor érdekes, ha a felhasználó a postafiók mérete után fizet a szolgáltatójának. De mégis rossz ötletnek tartom ezt a megoldást, mert ha vesszük például a ***CRM114***-et, amelyről 99,87% pontosságot említenek (ez durván 1 téves azonosítás 1000 levelenként), felmerülhet a kérdés, hogy mi van akkor, ha pont abban a levélben volt egy nagy összegű megrendelés? Higgye el az olvasó, ***Murphy*** törvénye szerint előbb-utóbb lesz ilyen. Hacsak az előbb említett termék nem a gyártó folyamatosan karbantartott adatbázisából dolgozik. Ebben az esetben azonban lényegesen kisebb a rugalmassága, hiszen egy uniformizált spam adatbázis sohasem veheti fel a versenyt a mi leveleinkből képzett adatbázis pontosságával. A ***clapf*** ezért (vírusfertőzés esetét kivéve) soha nem utasít el levelet, csak megjelöli, és a felhasználóra bízza, mit tesz vele. Az átlagos spam mérete jellemzően néhány kB – bár a csatolt képet tartalmazó spam-ek 40-100 kB körüli méretűek is lehetnek. Ezért a `max_message_size_to_filter` paraméter segítségével megadható egy méret, amely fölötti leveleket spamszűrés nélkül átengedi.

Feladat megoldva – vagy mégsem?

A figyelmes olvasónak bizonyára feltűnt az a diplomatikus megfogalmazás, hogy a ***clapf*** csak „nagy valószínűséggel" tudja eldönteni, hogy az adott levél spam vagy sem. Miért nem tudja biztosan? Bármely ***Bayesian*** szűrő minőségét két dolog határozza meg:

a) Milyen ügyesen bontja tokenekre az üzenetet,
b) a spam adatbázis minősége.

A spammer-ek sokszor ártatlannak látszó szavakat tesznek az üzenetbe (***poisoning***), amelyek valószínűleg nem szerepelnek a spam adatbázisban – sőt, ha egy mód van rá, akkor olyan szavakat, amelyek előfordul(hat)nak a felhasználó ham adatbázisában, mert ekkor sokkal kisebb lesz a levél összesített spam valószínűsége. A cikk írásakor kaptam egy olyan spam-et, amely az üzenet mellett tartalmazott egy részletet a ***„Tarzan New Yorkban"*** című műből. Ezért ma már a ***Bayesian*** spamszűrők csak a semleges középértéktől (0,5) leginkább eltérő 15-20 tokent vizsgálják meg, és ebből számítják ki az egész levél spam valószínűségét.

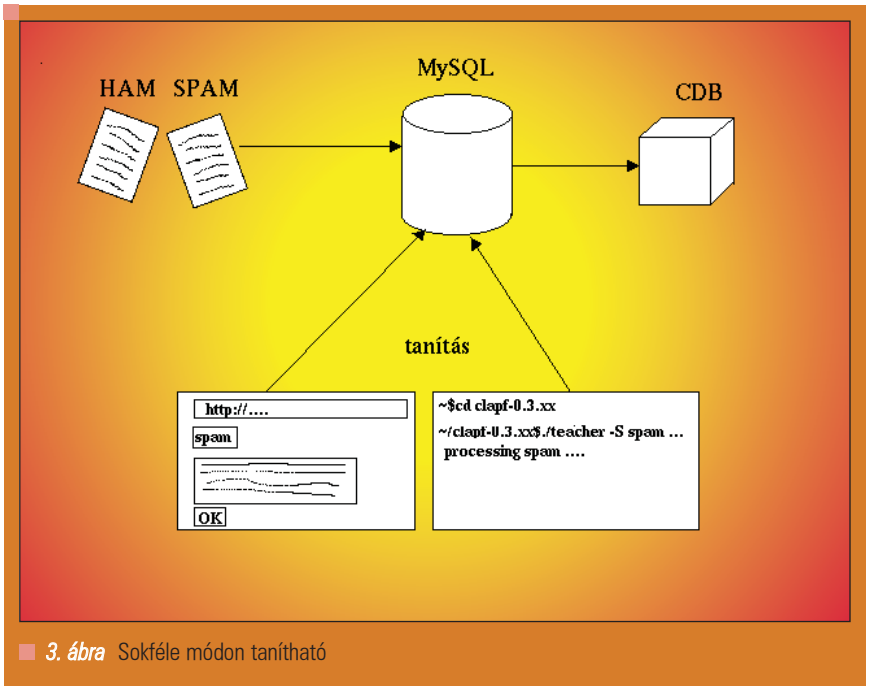


2. ábra Ezzel még a nagymamám is elbánna

Szóval ott jártam, hogy elkészült a spam adatbázis, minden rendben működött, de időnként mégiscsak becsúszott néhány spam. Az ilyen leveleket elemezve kiderült, hogy a vizsgált 15 token között bár voltak spam-re jellemző tokenek, de olyanok is, amelyek „ártatlannak" tűntek, és csökkentették az összesített valószínűséget, így a clapf által számított eredmény körülbelül 0,56-0,74 között volt, ami még nem spam, inkább bizonytalanságot jelez. A levelet tovább vizsgálva kiderült, hogy a top15 tokenen kívül volt még bőven spam-re jellemző token a levélben. Ezért módosítottam a ***clapf***-ot, hogy a semleges középértéktől nagyon eltérő összes tokent vegye be a számításba (`use_all_the_most_interesting_tokens=1`). A küszöbértéket nyilván nagyon magasra kellett állítanom, hogy ne lehessen ezzel megzavarni a számítást. Azonban időnként kaptam olyan spam-et is, amelyben az így kibővített top15 tokenek között óriási volt a spam tokenek aránya. Ezért egy új funkciót készítettem: ha itt a spam / összes token aránya nagyobb 0,9-nél (`spam_ratio_in_top10`), akkor a levelet spam-ként jelöli meg.

A ***DSPAM*** nyomán a ***clapf*** már nem csak önálló szavakat, de azok kombinációit (2 ill. 3 egymást követő tokenből képezi) is figyeli, mert ez növeli a ***Bayesian*** döntés pontosságát. A precízebb döntés érdekében a ***clapf*** már súlyozza a tokeneket és azok kombinációit aszerint, hogy hány elemből állnak. Minél specifikusabb, annál nagyobb súlyt kap, azaz egy 3 elemből álló kombináció többet nyom a latban, mint egy sima token. ***W. Yerazunis*** szerint a ***Bayesian*** elvvel van egy nagy baj: csak akkor érvényes, ha egymástól statisztikailag független eseményekre alkalmazzuk, azonban ez nem igaz levelek esetében. De még ezzel az alapvető logikai hibával együtt is rendkívül jól működik. A ***Bayesian*** szűrők pontosságának 99,9% az elméleti határa (➲ ***http://crm114.sourceforge.net/Plateau_Paper.html***), ami 1 tévesztést jelent 1000 levelenként, és ez azért nem rossz. De szerinte ez az arány tovább javítható a Markov diszkrimináció alkalmazásával. A legutolsó fejlesztői verzióban tesztelem ezt a változatot, ez ugyanis precízebb súlyozást tesz lehetővé.

Külön érdekesség a keleti spam, amely az én terminálomon sok esetben értelmezhetetlen, és sajnos nem tudok ezen a nyelven, hogy elmondhassam nekik, hogy ha ilyen karakterkészletet használnak, akkor még csak esélyük sincs, hogy tőlük vegyek ***Viagrát*** (vagy mit akarnak rám tukmálni). Ezt úgy küszöböltem ki, hogy készítettem egy `invalid_junk_characters[]` nevű tömböt, ahol több kínai, koreai jelet soroltam fel, amelyek tipikusan előfordulnak az ilyen és csak az ilyen levelekben. A ***clapf*** ezeket a karaktereket számolja, és egy határ után nem büntetőpontot ad, hanem az egész levelet spam-nek tekinti. Erre a problémára egyébként az az igazán frappáns megoldás, ha figyeljük az adott levél karakterkészletét, és ha az garantáltan értelmezhetetlen (például ***GB2312***, ***iso-2022-jp***, ***big5***, ***Windows-1251***), akkor azt eleve egy külön mappába gyűjtjük, amit én egy egyszerű ***maildrop*** szabállyal oldottam meg. Eddig csak spammer-ektől láttam, hogy – a levél elemzésével működő szűrőket megtévesztendő – akkor is ***base64***-kódolva küldik el az üzenetet a levél törzsében, ha annak típusa (text/plain) egyébként ezt nem tenné szükségessé. A ***clapf*** ezt képes dekódolni, ahogyan az ***UTF-8*** és ***Quoted-printable*** kódolt részeket is. Valószínűnek tartom, hogy a base64 kódolt, sima szöveges leveleket a jövőben automatikusan spam-nek fogom tekinteni. Az egyik felhasználó részéről felmerült egy olyan igény, hogy érdemes lenne vírusos levél vagy spam esetén értesíteni a feladót vagy a helyi postamestert. Azonban ezt nem tartom jó ötletnek, mert jelenleg a spam-ek tekintélyes részének hamisított a fejléce, másrészt a spam-ek tényleges küldői több esetben megfertőzött, esetleg feltört gépek, akiknek hiába is menne vissza egy panaszkodó levél, arról nem is beszélve, hogy kevés értelmét látom például 200 ***Zafi-B*** vírus esetén 200 vírusról értesítő levélnek ugyanazzal a szöveggel. Kérését mindenesetre teljesítettem, és a konfigurációs fájlban ez a funkció is aktiválható (alapértelmezésben kikapcsolt).



3. ábra Sokféle módon tanítható

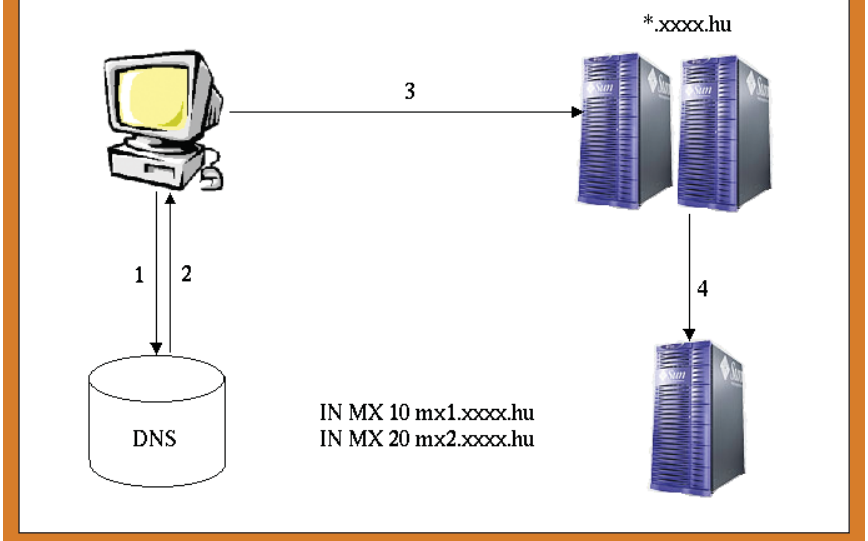
Hogyan tovább?

Egy időben néhány heurisztikus tesztet is implementáltam (például `Cc:` mező-ben szereplő címek száma, van-e benne IP-címmel megadott url, html tag-ek aránya, stb.). Sok spam-et megfogtam velük, azonban ezek könnyelmű beállítása megnövelte a téves pozitív riasztások esélyét. Ezért a 0.3.21-rc1-es verziótól kezdve a clapf már nem tartalmaz heurisztikát.

Aztán szembetalálkoztam a „jövő szemetével" (***Graham*** nevezte így), amely csak egy – viszonylag hosszú - URL-t tartalmazott (a ***http://uk.geocities.com/***... alatt valahol), ahol a spam tényleges üzenete olvasható. Mindezt pár véletlenszerű szóval fűszerezték meg, hátha sikerül még jobban összezavarni a ***Bayesian*** döntést. Egyelőre azonban nem akarom azzal komplikálni a ***clapf***-ot, hogy letöltöm az adott weboldalt, és annak is elemzem a tartalmát. Egy másik komplikált eset az, amikor a levél csak egyetlen képet tartalmaz, amit megnyitva olvasható a konkrét spam. Ez ellen úgy lehet védekezni, hogy egy képfeldolgozó alkalmazás segítségével kinyerjük a szöveget, és azt a megszokott módon elemezzük. Azonban a cikk írásakor nem találtam erre a célra megfelelő grafikus könyvtárat.

A ***clapf CDB*** formátumú token adatbázisból dolgozik. Azért esett erre a választásom, mert ezt találtam a legegyszerűbb megoldásnak, továbbá nem igényel külső relációs adatbázist – például ***Pgsql***, ***MySQL***) – amit külön felügyelni kell, csak olvasás módban nyitja meg a token adatbázist, így az nem hibásodik meg, nem kell zárolással foglalkozni, illetve rendkívül gyors, csak 2 diszk művelet, és megvan a keresett kulcs értéke. A jelenlegi körülbelül 7000 levélből (ham+spam) képezett token adatbázis mérete □29 MB, és körülbelül 602k rekord szerepel benne. A legutolsó fejlesztői változatban már 3 részre bontottam a token adatbázist, hogy minél gyorsabban le tudjam kérdezni a valószínűség értékeket. Hogy még kezelhetőbb legyen, a clapf ún. redukált adatbázist használ, amely nem tartalmazza azokat a tokeneket, amelyek csak egyszer szerepelnek vagy a ham vagy a spam tokenek között.

Egy hátránya azonban van a ***CDB*** formátumnak: nem frissíthető, márpedig téves azonosítás esetén tanítani kell. Ha módosítani akarjuk az adatbázist, akkor újra le kell generálni, ami akár pár percig is eltarthat. A ***doc/TRAINING*** fájlban azt fejtegetem, hogy az (lenne) a legegyszerűbb megoldás, ha a felhasználók elküldhetik a tévesen azonosított leveleket 2 belső email címre – az egyikre a tévesen spam-ként azonosított leveleket, a másikra a szűrőn átcsúszott spam-et, amelyből aztán rendszeres időközönként frissíteni lehet a központi adatbázist. Azonban ahány levelezőprogramot megnéztem eddig, annyiféle módon továbbítják a leveleket: az egyik csatolt fájlként, a másik beágyazott (inline) szövegként. Jelenleg azt a verziót preferálom, hogy



4. ábra clapf elosztott környezetben

a felhasználók egy spártai web oldalon egy űrlap segítségével taníthatják a *MySQL* adatbázisban szereplő tokeneket. Ebből pedig egy időzített feladat segítségével rövid idő alatt elkészíthető a *CDB* adatbázis.
Tervezem, hogy készíteni fogok egy programot, amelyet megfelelően felparaméterezve az adminisztrátor parancssorból is taníthatja a *MySQL* adatbázisban levő tokeneket. Ez megoldja a *Maildir* formátumú levelek importálását is. Lehet, hogy a jövőben közvetlenül *MySQL* adatbázisból fog dolgozni a *clapf*.
Az előbb vázolt megoldásban minden felhasználó egy közös adatbázison osztozik. Jobb megoldás azonban az, ha minden postafiókhoz tartozik egy – a felhasználó által karbantartható – token adatbázis, és a *clapf* mindig a megfelelőt használja. Ez azonban intenzívebb felhasználói közreműködést igényel, amire nem mindenki képes vagy hajlandó, ill. ehhez megfelelő diszk kapacitás is szükséges. Az egyes email címekhez (pontosabban felhasználói fiókokhoz) tartozó adatbázisok összerendelését tarthatjuk adatbázisban vagy akár *LDAP* kiszolgálón is.
Egy későbbi verzióban valamelyik megoldást implementálni fogom.
Aki pedig nem szerepel a címtárban, az használhatja az alapértelmezett központi adatbázist.
Egy web oldalon az adminisztrátor különféle statisztikákat fog majd tudni lekérdezni, például a spam/ham arányát, a feldolgozás átlagos idejét, különféle színes grafikonokat, stb. De ez a funkció a cikk írásakor még nincs kidolgozva.
Bár eddig folyton csak rosszakat mondtam a feketelistákról, bizonyos környezetben mégis jól használhatóak. Készítsünk egy csapda email címet, amit aztán mindenhol „reklámozunk" – csak ne a legitim partnereinknek. Ha erre a címre levél jön, az csak spam lehet, és a fejlécét elemezve, ki lehet nyerni belőle a feladó IP-címét, és azt egy helyi feketelistára tenni néhány órára. A *clapf* képes a bejövő levelek fejlécében szereplő `Received: from` sorok elemzésére, és ha az azokban talált IP-címek valamelyikét megtalálja az adatbázisban, akkor a levelet spam-ként értékeli (a *clapf* nem utasít el egyetlen levelet sem). Ezt a technikát nevezik oltásnak (*inoculation*).
Az is megoldható, hogy ne az *SMTP* szerverünkkel közvetlen kapcsolatba került feladót büntessük így, hanem keressük meg azt az IP-címet, ahonnan elindult a levél. Azonban a fejlécben szereplő útválasztási információ hamisítható, így fennáll a veszélye, hogy a feketelistánkra olyan címek kerülnek, amelyeket pedig nem akarnánk oda tenni. Ezért ezt változatot egyelőre nem támogatom, legalábbis amíg nem készítek egy olyan kódot, ami a fejléc információk hamisítását képes detektálni. Hamisított fejléc esetén viszont egészen biztosak lehetünk benne, hogy a feladó szándékai nem tiszták, így bátran megjelölhetjük az ilyen leveleket.

Egy *cron* héjprogram is szükséges az oltás teljesen automatizált működéséhez, amely rendszeres időközönként törli a lejárt bejegyzéseket. Az oltás az `--enable-blackhole` opcióval aktiválható. Ebben az esetben ajánlott a *clapf* konfigurációs fájl jogosultságát 640-re állítani, nehogy illetéktelenek hozzáférjenek egy adatbázis fiókhoz.
Előző írásaimban több konkrét spamszűrőn mutattam be a leggyakoribb védekezési elveket. Előfordulhat azonban, hogy egy vállalatnál nincs erőforrás erre a feladatra. Ma már azonban vannak olyan szolgáltatók, akik vállalják, hogy egy cég teljes bejövő levelezését átvizsgálják vírus és spam tekintetében. Ezt általában úgy teszik, hogy az adott tartomány *MX* rekordjait az ő gépeikre állítják be. Így a megbízó cég összes levele a szolgáltató szerverfarmjára megy. Itt átvizsgálják a leveleket, tartalmaz-e vírust vagy spam-et, majd ennek eredményét beillesztik a levélbe, és továbbítják a cég levelező szerverei felé. A *clapf* egy ilyen környezetben is képes helyt állni.

A számok

Egy 12 napos időszak alatt 445 levelet kaptam, megfogott 119 spam-et, de egy azonban átcsúszott a szűrőn, így a pontossága 99.77% volt. A kategorizálás az én átlagos (<10kB) méretű leveleimnél 20-30 ms körüli időt vesz igénybe. A pontosságot illetően el kell mondanom, hogy valójában több levelet kaptam, például különféle listákról, de azokkal nem tanítottam az adatbázist, így azokat nem is vettem figyelembe a pontosság számításakor.
Angol levelek esetén időnként előfordult téves pozitív azonosítás, ami a korábban említett heurisztikus változók finomabb hangolásával (talán) elkerülhető lett volna. A heurisztikus elemzés eltávolítása óta azonban nem találkoztam ilyen hibával. Mire az olvasó idáig ért, valószínűleg már további finomításokat, funkciókat építettem a szűrőbe.

Sütő János
(jsuto@freemail.hu)
1997 óta használ Slackware Linux-ot. Szabadidejében a postfix clapf nevű vírus- és spamszűrőjét polírozza.

Száguldás, szerelem

Vajon akad-e bárki, aki ne szeretne exkluzív és veszedelmesen gyors autókat vezetni, mindenféle kockázat na és a „kapcsolódó szankciók” nélkül? Kötve hiszem, hogy bárki ellenkezne, ha lehetősége nyílna rá. Személy szerint még egy hétköznapi autó volánja mögött is igen nehezen tudom megállni, hogy ne nyomjam tövig a gázt...

Hiába, sokunkat vonz a sebesség. Esetemben elég régi szerelem ez. Tisztán emlékszem, amikor szüleimtől 1986-ban megkaptam nyolcadik születésnapomra a „nagy meglepetést”, egy ***8 bites*** számítógépet. Rövidesen le sem lehetett „vakarni” a képernyőről: amikor csak tehettem, kedvenc programjaimat nyúztam. Akkortájt leginkább az autóversenyeket idéző játékok kötöttek le. Ez a vonzódás a mai napig megmaradt. Bár az igazi vezetés már jobban foglalkoztat, néhanapján linuxos gépemen jelentős időt töltök a ***Need For Speed*** sorozat valamelyik korai tagjával (a ***Win32 wrapper***-ek kompatibilitási rétegére hagyatkozva), vagy valamely „tisztán” linuxos programmal. Az utánzott környezetet igénylő játékokról nem akarok írni, így most azt fogom megmutatni, milyen natív megoldások közül választhat az a játékos, aki versenyezni szeretne.

Torcs

A ***Linux*** platform egyik közismert autós játéka a ***TORCS*** névre hallgat. Igen lényegre törő felépítésével saját kategóriájában kiforrott megvalósításnak számít. A program a ➲ ***http://torcs.sourceforge.net*** oldalon elérhető ***GPL*** licenszű forráskód és független bináris formájában egyaránt. Forrásának felépítése a szokásos módon a kibontás után kiadott

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

parancsokkal történik (mindezt természetesen rootként kell megtennünk). A kód csak akkor használható, ha a beállítószkript minden függőséget megtalált. Legfontosabb, hogy a ***3D*** megjelenés miatt ***OpenGLt*** támogató grafikus kártya megléte szükséges, melyet valamely ***GLUT*** csomaggal is meg kell támogatnunk. Aki nem szeretne időt tölteni a fordítással, választhatja az előre lefordított ***i686*** binárist is, rendszere csomagkezelőjének megfelelő formában, esetleg ***Loki*** alapú telepítőbe ágyazva. A programkód moduláris felépítésű, így az autók és pályák száma akár fordítás után is könnyedén bővíthető: mindössze a telepítési út megfelelő ***/tracks*** és ***/cars*** almappájába kell mozgatni a kívánt autók és egyéb komponensek állományait. Felhasználóként, terminálra gépelt `torcs` paranccsal indíthatjuk a játékot, mely kulturált módon, minden személyhez kötődő konfigurációs beállítást és eredménytáblát a ***/home/$user/.torcs*** úton tárol. Nézzük meg közelebbről, mit várhatunk a ***TORCS***-tól!

Számítógép által vezetett ellenfelekkel szemben élhetünk át országúti és körversenyeket, valamint rallycrosshoz hasonló megmérettetéseket. Alaphelyzetben megközelítőleg tizenöt pályánk van, és nagyjából ugyanennyi autónk. Akár kieséses rendszerben is játszhatunk. Ha szükségünk van rá, akkor több játékos módot is választhatunk, és nem mellékesen osztott képernyőn is lehetőségünk van kergetni egymást. Utóbbi mód a dicső múltat idézi: rendkívül szórakoztató,

■ *1. ábra* A TORCS világa

■ *2. ábra* A Racer megvalósítása

ahogyan több játékos vetekszik egymással egy gép előtt ülve. Az autók viselkedését illetően a program fizikája által megengedett ***arcade*** vonalra kell számítanunk, melyben a játékmódtól függően autónk állapotára és a tank töltöttségi állapotára is ügyelnünk kell.

A ***TORCS*** grafikailag „középúton" jár, az autók és pályák megvalósítása javarészt ***Need For Speed II*** színvonalú. Érdekes, hogy amennyiben játékos beállítottságod mellett egyaránt rendelkezel „művészi vénával" és mérhető türelemmel, akkor a géppark egyszerűbb autóinak festésén könnyen segíthetsz: a telepítési úton fellelhető ****.rgb*** formájú textúrák a ***GIMP*** által használhatóak és módosíthatóak. A fentebb említett honlap látogatóinak száma mára már milliós nagyságrendben mérhető. Nem véletlenül: ***GPL*** licenszű kódra épülő autóversenyek között aligha találni ennél letisztultabb darabot. Robotjainak fejlesztését a ➲ ***http://www.berniw.org/trb*** címen lehet követni, pályaszerkesztőt pedig a ➲ ***http://katergo.rege.org/projects/trackeditor*** oldalon érdemes keresni.

Racer

A ***Racer*** névre hallgató projekt egy holland fejlesztői bázisú, nyitott forrású autóverseny. A ***TORCS***-tól eltérően sokkal inkább szimulátoros tartalommal rendelkezik, ezen felül grafikai kivitelezése sokszor már mérnöki pontosságú. Forráskódja a ➲ ***http://www.racer.nl*** címen érhető el. Ennek a lefordítása meglehetősen nehéz, rengeteg függőséggel kell számolnunk. Legfontosabb a ***3D*** képességű grafikus hardver, amely itt is alapkövetelmény, emellett szükség lesz megfelelő ***GLUT*** csomagra és ***FMod*** hangkönyvtárra is. A fordítás aztán már egy `make` paranccsal lefut, amit értelemszerűen a ***Racer*** forrásának gyökerében kell kiadnunk rootként.

Ennek hatására a ***/src*** úton létrejön egy indítható állomány ***racer.bin*** néven, ami mellé megszabott struktúrában kell bemásolnunk a letöltött adatcsomagokat. Amennyiben nem szeretnénk vesződni ezekkel a műveletekkel, érdemes meglátogatni a ➲ ***http://liflg.org*** címen elérhető ***LIFLG*** csapatot, mely a nagy sikerű játékot disztribúciótól független natív binárisként, ***Loki*** - alapú telepítőben terjeszti. Akármelyik megoldást is választjuk, fel kell rá készülnünk, hogy a kód igen válogatós grafikai hátterünket illetően. Bizonyos ***ATI*** kártyákon nem indul, bizonyos ***nVidia*** meghajtók esetén képe hibázik. Érdemes megjegyezni, hogy a projekt még mindig béta állapotú, így ezektől a nehézségektől várhatóan meg fog szabadulni a közeljövőben, bár a fejlesztése igencsak lassan halad.

A ***Racer*** versenymódok tekintetében országúti és körversenyeket ismer. Egy rövid gondolat erejéig visszakanyarodnék a szimulátor jelleghez: néhányan tudják, kedvenc autóim a ***Ford Mustang*** szériák korai darabjai. Ha valaki ért a kocsikhoz, akkor tudja: az esetek többségében ez egy olyan „öblös" gép, mellyel tapasztalatlan vezető elsőre el sem tud indulni. A könnyem majd kicsordult az örömtől, amikor ezt a képernyő előtt viszont tapasztaltam. A gyönyörű, virtuális ***Mustangom*** csupán harmadik sebességben volt biztonsággal indítható - nagy fröccsel „egyesben" azonnal kipörgött, és füstölgő gumikkal jelezte, hogy finomabban kell bánnom a motortérben „tomboló lovakkal". A gépek természetesen típusuknak megfelelő váltási időket kaptak, emellett (szintén élethűen) rossz fokozatba kapcsolva a motor néha le is fulladhat. Erre felkészülve a program természetesen lehetőséget ad önindításra. Az autók modellezői általában törekednek az élethű motorhangra is, így erre sem lehet panaszunk. További jó hír, hogy kötött kameranézet itt nincs, a billentyűzetünk numerikus részével állíthatjuk be a nekünk megfelelő látószöget és távolságot (akár az autó belsejébe is pozicionálhatunk!). A tetszetős dolgok ellenére sajnos a program egyelőre csak hálózati módban említhető teljes értékű játékként, mivel a némely ***build*** verzióban megvalósított gépi játékosok igen gyengék. A játék felhasználóként, terminálra gépelt `racer` paranccsal indítható (esetleg a fordítás által létrejött bináris állománnyal).

A ***Racer*** szerkezete moduláris megvalósítású, melynek ügyén egy kiemelten fontos dolgot meg kell jegyeznem. A pályákat és modelleket leíró fájlok nevében sokszor logikától mentesen és vegyesen szerepelnek a kis / nagy betűk. ***Win32*** alatt ez valójában jelentéktelen hiányosság, ***Linux*** alatt viszont már okozhat problémát. Így, ha egy frissen letöltött modellen textúrák helyett csupán piros poligonokat látsz,

akkor a pályát, modellt leíró ***.ini** fájl szerint kell alakítanod az adatcsomagok neveit. A program részletes finomhangolása a ***racer.ini, modeller.ini*** állományok kézi szerkesztésével tehető meg, mivel a menüszerkezet csak a legszükségesebb paramétereket engedi megváltoztatni. Ha valaki nem boldogul az összetett szerkezettel, szívesen segítek megértésében, akár elektronikus levélben is – esetleg kicserélhetjük ez irányú tapasztalatainkat.

3. ábra Íme, a Trigger

Trigger

Az előző két kódtól eltérően a ***Trigger*** már nem aktív, mivel a legfőbb fejlesztő munkája nemrég olyannyira megtetszett az egyik nagynevű konzolgyártónak, hogy le is szerződtette saját köreibe. A „félbehagyott” játék ennek ellenére fellelhető az eredeti ➲ ***http://www.positro.net/trigger*** oldalon, ***GPL*** forrásként és futtatható kódként egyaránt. Igen egyszerű és lényegre törő projektről van szó, melyben néhány pályán versenyezhetünk az idő ellenében, szimulátoros viselkedésű autókkal. A program felépítése szintén moduláris, így adatok tekintetében könnyen bővíthető. Sajnos sem gépi ellenféllel, sem hálózati lehetőséggel nem fogunk találkozni, mivel ilyen képessége nincs implementálva (a rallyszerű játékmód egyébként ezt nem is igazán igényli). Mivel a bináris kód bő fél évvel ezelőtti állapotot mutat, így érdemes a forráskód lefordítására időt szakítani, hiszen a hivatalos, előre fordított ***Trigger*** meglehetősen régi rendszerkomponensekre támaszkodva lett felépítve. A játékot lefordítani ***Jam*** segítségével lehet, rootként kiadott `./configure`, `jam` parancsokkal. Fő függőségei: ***PhysFS***, ***SDL***, ***OpenGL***, ***OpenAL***, (***Jam***). A projekt fő mappájában található (illetve az ott létrejött) triggerállomány segítségével indítható. A közvetlenül emellett fellelhető ***trigger.conf*** fájlt mindenképpen érdemes átnézni még első indítás előtt és kikapcsolni benne a jelentőséggel nem bíró grafikai opciókat, valamint beállítani a kívánt felbontást. Mielőtt a programozási munkák félbeszakadtak volna, lázasan dolgoztak az új verzió fizikai modellezését megvalósító főkódon, mely (szintén félbehagyott állapotában) megtalálható az említett honlapon.

4. ábra Az UnWheel „terepmódja”

UnWheel

Mire ez a cikk megjelenik, talán már „lement” az ***UT2004*** írásom, melyben ezt az érdekes ***MOD*** - ot már megemlítettem. Az ***UnWheel*** egy olyan modifikáció, mellyel a híres ***FPS*** egy autóversennyé szelídíthető. Persze mindez ennyire nem egyszerű, hiszen a kód erőssége nem elsősorban az autóverseny jelleg, hanem az ezen belüli csapatjáték módok, mint a területbirtoklás, zászlólopás, „nyúlfuttatás”. Ennek ellenére „mezei”, ***arcade*** verseny is indítható általa, akár rally pályákon, speciális autókkal.

Apropó autók: itt a ***FORD*** „puttonyostól” kezdve az oldalkocsis motoron át akár ***CAT*** dömper is vezethető. Nagyjából huszonöt járművet választhatunk tíz játékmóddal, cirka harminc pályával keverve. Ugye, nem is hangzik rosszul?
Személy szerint a ***Monster Trucks*** pályákat szeretem leginkább, a hozzá illő járgányokkal együtt. Egy szépséghibája azért akad az egésznek, mégpedig a gépi játékosok terén. Valójában csak hálózati módban teljes értékű a program, hiszen egyedül játszva a számítógép vezette ellenfeleink igen szerény képességűek. Az ***UnWheel*** legkönnyebben a ➲ ***http://liflg.org*** oldalról letöltött, ***Loki*** – alapú telepítőbe ágyazottan állítható üzembe. Mindössze annyi dolgunk akad, hogy a letöltött ***unwheel.verzió.run*** telepítőállományt elindítjuk, miután a rendszerünkben lévő ***UT2004*** játékmotorját frissítettük az elérhető legmagasabb verzióra. Az autós ***MOD*** az ***Unreal*** menüiből indítható. Hardverigénye nagyjából az „anyaprogrammal” megegyezően magas.
A téma iránt érdeklődőknek mindenképpen érdemes keresgélniük a világhálón, hiszen létezik még néhány életképes megoldás. Ezek közül a legfontosabb talán az egyelőre gyerekcipőben járó ***VDrift***. Az igazán komoly „egy játékos” móddal rendelkező autós projekteket jelenleg sajnos nem írják át ***Linuxra***. A ***Need For Speed*** rajongóknak viszont jó hír lehet, hogy a ***Cedega 4.x*** verziótól kezdve simán megbirkózik akár az ***UnderGround*** részekkel is, aki pedig a rallyt kedveli, az a ***Rally Trophy***-t is életre keltheti az utánzott környezetben.
Mindenkinek kellemes száguldozást, és tartalmas kikapcsolódást kívánok!

Kovács Zsolt (kovi@linuxforum.hu)

Quake fanatikus. Négy éve a debreceni linuxosok egyike. Töretlenül hisz a Slackware terjesztésben.

Kiadványszerkesztés Linux alatt – Scribus
PDF fájl készítése (3. rész)

A számítógépes tördelőprogramok megjelenésének idején sokan gondolhatták: „Mivé lett ez a világ?! Mi lesz a tradíciókkal?! A nyomdászmesterségekkel?! A betűk kézzel történő kirakosgatását fel tudja váltani egy szoftver?” Azóta kiderült, igen, fel tudja váltani és a tradíciókat sem kell elfeledni, akárcsak a mesterségeket. Kényelmesen elférnek egymás mellett.

A *PDF* formátum térnyerésekor is valami hasonló játszódhatott le. A kézzelfogható könyv, újság helyett egy monitoron olvasható digitális állományt „vehetünk kézbe” Mint azóta kiderült, nem szükséges a papír alapú kiadványokat mellőzni, ez a formátum inkább kiegészíti ott, ahol az már nem használható fizikai korlátai miatt. Ez a hely pedig az Internet, de ez csak egy lehetőség a sok közül.

Röviden a PDF szabványról

A dokumentum formátum körülbelül 10 éve kezdte meg hódító útját és egyre többen éreznek rá az ízére, veszik észre hasznosságát. A nyomdai előkészítésben történő komolyabb szerepvállalását 4-5 éve kezdte meg. E téren addig jutott, hogy elég egy jól megszerkesztett PDF dokumentummal lerohanni a közelben lévő sarki digitális nyomdába és akár indulhat is a nyomtatás.
A *PDF* egyik előnye a hordozhatósága, a *PostScripttől* eltérően sokkal kisebb fájlméretet lehet elérni és ez teszi lehetővé az Interneten keresztüli gyors átküldését. Valamint az egyre bővülő funkcióit is érdemes megemlíteni, egy egyszerűbb áttekintést foglaltam össze az *1. táblázatban*.
A *Scribus* egyik erősségeként a *PDF* támogatást szokták emlegetni (elsőként támogatta a *PDF/X-3* szabványt), ami valóban figyelemreméltó.

1. táblázat *A Typography menü tartalma*

PDF 1.3	Az első nyomdabarát formátum, színprofil támogatással (ICC), Javascriptek beépítésével, interaktivitást lehetővé tevő kiegészítőkkel.
PDF 1.4	Ekkor került bevezetésre az áttetszőségek kezelése.
PDF/X-3	Egy új lehetőség, a CMYK mellett az RGB színek és színterek használata is engedélyezett. A PDF 1.3-at vették a szabvány alapjául.
PDF 1.5	Jelenleg ez a legújabb verzió, a rétegkezelés időszámításának kezdete.
Automatic Line Spacing	Automatikus sorköz

A Preflight Verifier/Elővizsgáló menü

A készítendő *PostScript* és *PDF* dokumentumok (hiba)vizsgálatának finomságát hangolhatjuk be. A menücím alatti legördülő menüből választhatjuk ki azt a dokumentumszabvány fajtát, amelyre vonatkozó ellenőrzéseket akarjuk finomítani. A használni kívánt ellenőrzést a leírása előtti kis négyzetbe kattintott pipával jelezhetjük a *Scribus* felé.
Mindig engedélyezzük a hibakeresést, ezért az *Ignore all errors*-t hagyjuk üresen és ne vegyünk tudomást a létezéséről. Természetesen a hibakeresés lelassítja a kívánt dokumentum elkészítését, viszont a jó paraméterezésnek köszönhetően nem lesznek kompatibilitási gondjaink.

- *Automatic check before printing or exporting*: erősen ajánlott a használata. Így nem érhetnek meglepetések nyomtatás vagy exportálás (ez még a kisebbik baj) közben/után. Nincs annál szebb, amikor egy többször tíz oldalas dokumentum kinyomtatásakor azt vesszük észre, hogy a fejlécekbe ágyazott díszítő ábra/kép nem került kinyomtatásra.
- *Check for missing glyphs*: a könnyebb azonosítás érdekében a hiányzó speciális karaktereket egy vörös négyzettel helyettesíti a szövegben. Ezeket kell kicserélni a megfelelő karakterekre.
- *Check for objects not on a page*: az oldalon kívül elhelyezkedő objektumokra hívja fel a figyelmet.

Az oldalról lelógó objektumot bizonyos mértékig figyelmen kívül hagyja, a lelógó részét levágja. Az *1. ábrán* a ***Scribus***-ban készült kilógó objektum, a *2. ábrán* az ***Acrobat Readerben*** megnyitott, kész ***PDF*** dokumentum „megnyírt" objektuma látható.

- ***Check for overflow in text frames***: a kiadványok leglényegesebb eleme döntő többségben a szöveg. Ha valamelyik szövegdoboz a benne található szöveg formázása után kicsinek bizonyul, akkor egy kis vörös „X"-elt négyzettel jelzi, hogy a szöveg túlfolyt. Erre a jelentős hibára figyelmeztet ez az opció.

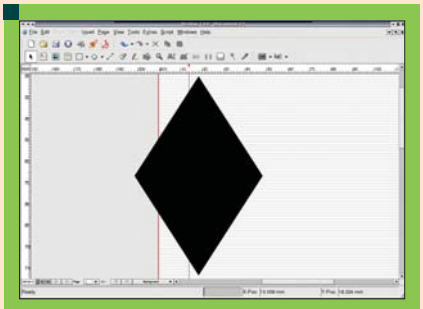
- ***Check for transparencies used***: az áttetszőség csak a ***PDF 1.4***-től támogatott funkció. A ***Properties/Colors/Opacity*** értékénél állítható az objektum áttetszősége.

- ***Check for missing images***: ez a már fentebb említett hiányosságot jelzi. Ha talál az ellenőrzés során egy üresen maradt képdobozt, akkor azt azonnal közli. A *3. ábrán* egy ilyen hibajelzésre megjelenő ablakot láthatunk.
  Az ablakon látható az éppen használt dokumentum ellenőrzési profilja (***PDF 1.4***) – amit a legördülő menüvel át is állíthatunk –, a dokumentum objektumai, összetevői (***Items***) és a hozzájuk kapcsolódó ellenőrzés eredménye (***Problems***). Mint az látható az üres képdobozra azonnal válaszolt is egy ***Missing Image*** üzenettel. Ha rákattintunk az objektum nevére (***Image0***), akkor a nézetet a hiba előfordulási helyére teszi át és javíthatjuk a hibát, aminek ellenőrzését automatikusan el is végzi. Lehetőségünk van továbbá a hiba figyelmen kívül hagyására (***Ignore Errors*** gomb), és jöhet a következő lépés: a ***PDF*** fájl megírása.

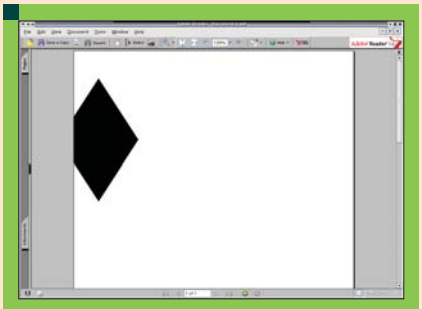
- ***Check image resolution***: a képek felbontásának és azok legkisebb értékének ellenőrzése illetve meghatározása. A képfelbontásokat mindig az adott célra kell megválasztani. Számítógépen való nézegetésre, Internetre elegendő a 72-150dpi közötti érték

2. táblázat *A Preflight Verifier/Elővizsgáló menü tartalma*

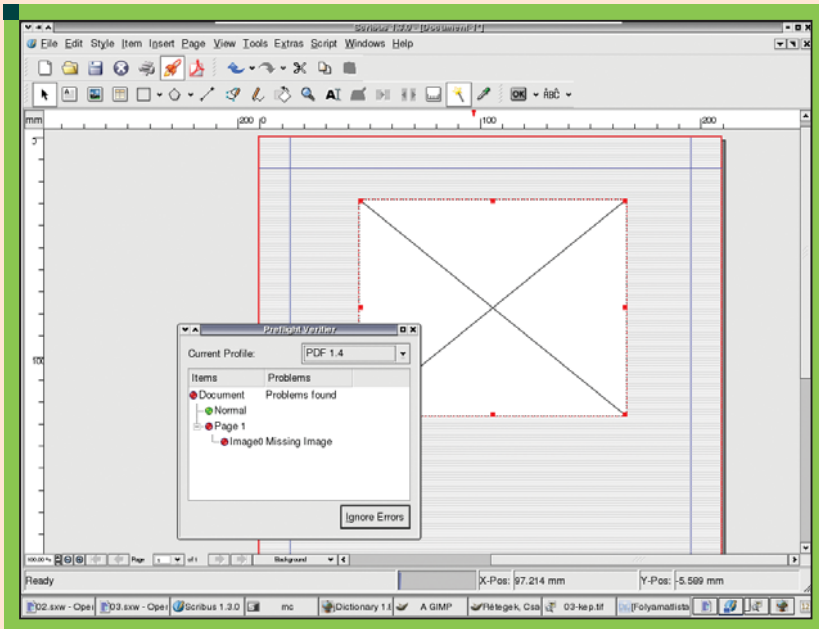
Ignore all errors	Az összes hiba figyelmen kívül hagyása
Automatic check before printing or exporting	Automatikus ellenőrzés nyomtatás vagy exportálás/mentés előtt
Check for missing glyphs	Speciális karakterek meglétének ellenőrzése
Check for objects not on a page	Az oldalon kívül elhelyezkedő objektumok ellenőrzése
Check for overflow in text frames	Túlfolyó szöveg ellenőrzése a szövegdobozban
Check for transparencies used	Áttetszőségek ellenőrzése
Check for missing images	Hiányzó képek ellenőrzése
Check image resolution	Képfelbontás ellenőrzése
Lowest allowed resolution	A legkisebb lehetséges felbontás
Check for placed PDF Files	Behelyezett PDF fájlok ellenőrzése
Check for PDF Annotations and Fields	PDF jegyzetek és mezők ellenőrzése



1. ábra A Scribus-ban készült még kilógó objektum



2. ábra Az Acrobat Readerben megnyitott már vágott objektum



3. ábra Az a bizonyos hibajelző ablak

3. táblázat *C-C-T-F*

C	littlecms-sel fordított változat (Color Management Support – Színkezelés Támogatás)
C	CUPS nyomtató kezelés
T	TIFF képek támogatása
F	Fontconfig támogatása

(természetesen, ha komolyabb céllal készül a dokumentum, és számít a képek minősége, akkor ettől el lehet térni). Nyomtatásra készülő PDF esetén a szürkeárnyalatos és színes képeket már eleve 300dpi-s felbontáson kell a dokumentumba illeszteni, tehát érdemes ilyenkor 300dpi-t beállítani az alsó határhoz.

- ***Check for placed PDF Files***: behelyezett ***PDF*** fájlokat ellenőrzi.
- ***Check for PDF Annotations and Fields***: jegyzetek (szöveg, link, stb.) és mezők ellenőrzése.

A Color Management/Színkezelés menü

A színkezelés biztosítja, hogy amit a monitoron munka közben látunk, ugyanazt kapjuk kinyomtatott formában is, a színekre vonatkozóan.
A ***Scribus***-t fel kell készíteni a profilok kezelésére. Vessünk egy gyors pillantást a ***Help/About Scribus*** almenüre és a felugró ablakon a ***Build-ID*** dátumát követően a ***C-C-T-F*** betűnégyesnek kell szerepelnie. A betűk jelentése a *3. táblázatban* olvasható.
A téma tekintetében a ***littlecms*** megléte a legfontosabb. A ***CMS (Color Management System)*** feladata a színterek közötti átalakítás elvégzése. Ehhez szükség lesz egy ***RGB*** (a monitorhoz) és legalább egy ***CMYK*** (a nyomtatáshoz) profilra, ami az adott eszköz színterét írja le. Az ***Adobe*** cég honlapjáról letölthető egy ***ICC*** profilgyűjtemény (a nyomtatáshoz), ami tartalmazza a legelterjedtebb, leginkább használatban lévő profilokat.
A monitorhoz legjobb a saját gyártó által készített profil beüzemelése.
A fájlokat a ***General*** menüben meghatározott helyre bemásolva tehetjük használhatóvá.

4. táblázat *A Color Management/Színkezelés menü tartalma*

Activate Color Management	Színkezelés engedélyezése
System Profiles	A szerkesztő rendszerre jellemző profilok
Pictures	A képeket kezelő alapértelmezett színprofil
Solid Colors	A homogénszíneket kezelő alapértelmezett színprofil
Monitor	A monitort kezelő színprofil
Printer	A nyomtatást kezelő színprofil
Rendering Intents	A leképzés módja
Monitor	Monitoron
Printer	Nyomtatásban
Simulate Printer on the Screen	„Soft Proof” A monitoron a nyomtatásnak megfelelő színképzés lesz látható
Mark Colors out of Gamut	A nyomtatható színtartományon kívül eső színeket mutatja

- ***Pictures***: a képdobozokba betöltött képek alapértelmezett színprofilja.
- ***Solid Colors***: a képdobozokon kívül előforduló homogénszínek színprofilja.
- ***Monitor***: a monitor gyártója által készített színprofil. Ez egy speciálisan erre a célra készített profilfájl, ezért ha lehet, ne használjunk más profilt.
- ***Printer***: a nyomtatáshoz szükséges színprofil. Ez is egy speciális profilfájl akárcsak a monitoré.
- ***Rendering Intents***: a leképzés módja. Az alábbi négy lehetőség közül lehet választani:
- ***Absolute Colorimetric***: a nyomtatható tartományon kívülre eső színt a hozzá legközelebb álló már nyomtatható színárnyalattal pótolja. Hátránya, hogy telítettebb és esetenként világosabb színeket kapunk. A színtartományok között látható az összefüggéstelenség, valamint a szürke tartomány elszíneződése. Ajánlott terület: proof készítés.

A következő három eljárás alapjaiban megegyezik: az eredeti fehéret az eredménynek számító fehérként értelmezi, és a többi színt ennek megfelelően állítja be.

- ***Relative Colorimetric***: teljesen ugyanúgy képezi a színeket, mint az ***Absolute Colorimetric***, egy eltéréssel, a leképzést a fehérhez igazítva végzi. Ajánlott terület: vektoros rajzok, pasztellszíneket tartalmazó fotók.
- ***Perceptual***: eredeti színeket belepréseli a céltartományba. A színek telítetlenebbek lesznek, de megtartja az árnyalatok, színterek közötti átmenetet. Ajánlott terület: fényképek, különleges színeket tartalmazó vektoros rajzok.
- ***Saturation***: az eredeti kép elsődleges telített színeit a céltartomány telített színeinek felelteti meg. Élénk, eleven színeket ad, megtartja az árnyalatok, színterek közötti átmenetet. Ajánlott terület: „csillogó-villogó” grafikák.
- ***Simulate Printer on the Screen***: ezzel a funkcióval „házi proof”-ot készíthetünk a saját monitorunkon. A proof készítés lényege, hogy a végleges nyomtatás előtt egy próbanyomatot kapjunk a kinyomtatandó anyag valamely (például színügyileg „rázósabb”) oldaláról. Így még időben javíthatóak a hibák.
- ***Mark Colors out of Gamut***: a jelzett színek a nyomtatásban sötétebb, világosabb vagy éppen más

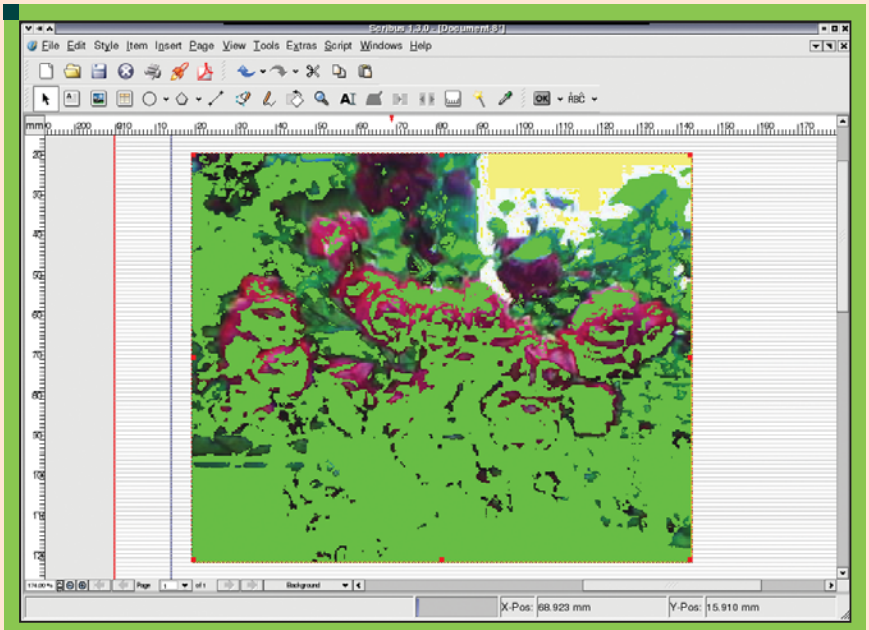
színtelítettséggel jelennek meg (*4. ábra*). Nem adják vissza az eredeti kép színességét.

A PDF Export menü

A számunkra legmegfelelőbb, legtöbbször használt *PDF* verzió beállítási lehetőségeit ezen a négy almenün tudjuk elvégezni. A munka során lehetőségünk adódik a ***File/Document Setup*** menüben a dokumentumra jellemző beállítási tulajdonságok finomítására. De ezt a ***Save as PDF*** ikonra kattintva megjelenő ismerős ablakon is elvégezhetjük.

General/Általános almenü

- ***Export Range***: a dokumentum exportálási tartományát állíthatjuk. Ez lehet a teljes dokumentum (***All Pages***) vagy egy bizonyos szakasza, oldalai (***Choose Pages***).
- ***Rotation***: a kimenetre kerülő oldalakat elforgatja a megadott szögnek megfelelően.
- ***Mirror Page(s) horizontally/vertically***: az oldalakat tükrözi vízszintesen vagy függőlegesen vagy akár mindkét irányba egyszerre. Ennek hasznát lehet venni, ha valaki öntapadós matrica hordozójára szeretne nyomtatni és ahhoz készít PDF-et.
- ***File Options***: a fájl verzióját (***Compatibility: 1.3, 1.4, 1.5***) és egyéb jellemzőit állíthatjuk be. A ***PDF/X-3***-at csak a készülő dokumentumban választhatjuk!
- ***Binding***: a kötés a lap melyik szélére kerüljön. Ez nálunk a baloldal, a jobboldali kötést – például – a japánoknál alkalmazzák, ahol „visszafelé” lapozzák a könyveket.
- ***Generate Thumbnails***: megkönnyíti a kész *PDF* dokumentumban való tájékozódást.
- ***Save Linked Text Frames as PDF Articles***: láncolt szövegdobozok mentése *PDF*-ként
- ***Include Bookmarks***: könyvjelzők beágyazása, szintén megkönnyíti a *PDF*-ben való közlekedést.
- ***Include Layers***: a dokumentumban használt rétegek beágyazása. A kész dokumentumban a beágyazott rétegeket a *PDF* olvasóval lehet ki- és bekapcsolni.



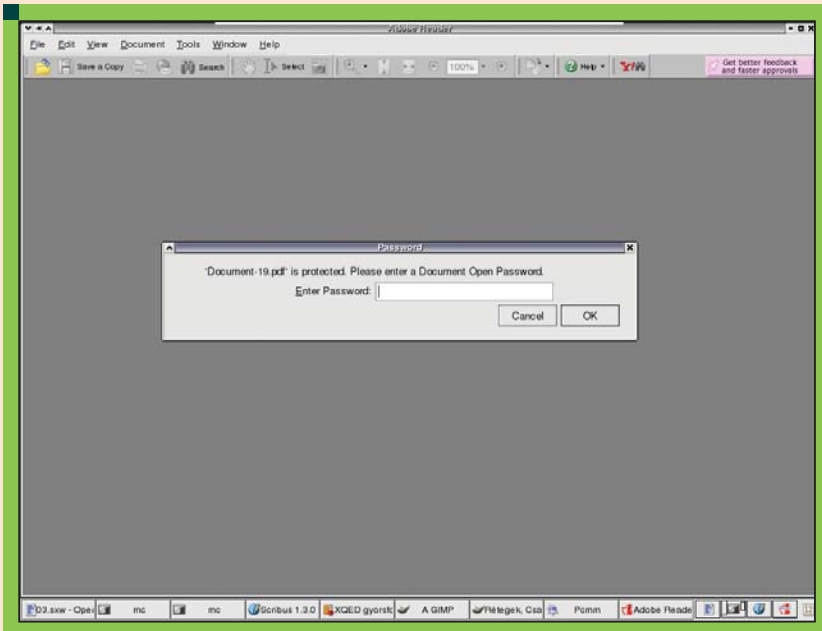
4. ábra Így fest egy virágoskert, ha gond van a színekkel (out of Gamut)

5. táblázat ***A General/Általános almenü tartalma***

Export Range	Exportálási tartomány
All Pages	Összes oldal
Choose Pages	Oldalak kiválasztása
Rotation	Elforgatás (0°, 90°, 180°, 270°)
Mirror Page(s) horizontally/vertically	Az oldal(ak) tükrözése vízszintesen/függőlegesen
File Options	Fájl beállítások
Compatibility	Kompatibilitás (PDF 1.3, 1.4, 1.5)
Binding (Left, Right)	A kötés helye (bal vagy jobb oldalt)
Generate Thumbnails	Bélyegképek létrehozása
Save Linked Text Frames as PDF Articles	Láncolt szövegdobozok mentése PDF-ként
Include Bookmarks	Könyvjelzők beágyazása
Include Layers	Rétegek beágyazása (csak PDF 1.5)
Resolution for EPS Graphics	Szöveges és vektoros grafikák képfelbontása
Compress Text and Vector Graphics	Szöveges és vektoros grafikák tömörítése
Image Settings	Képekre vonatkozó beállítások
Method (Automatic, JPEG, Zip, None)	Képtömörítési eljárás (Automata, JPEG, Zip, Nincs)
Quality (Maximum, High, Medium, Low, Minimum)	Tömörítési minőség (95%, 85%, 75%, 50%, 25%)
Resample Images to	Képek felbontásának átméretezése

6. táblázat *A Security/Biztonság almenü tartalma*

Use Encryption	Titkosítás használata
Passwords	Jelszavak
Owner	Tulajdonos jelszava
User	Felhasználó jelszava
Settings	Részletek beállítása
Allow Printing the Document	A dokumentum nyomtatásának engedélyezése
Allow Changing the Document	A dokumentum megváltoztatásának engedélyezése
Allow Copying Text and Graphics	Kép és szöveg másolásának engedélyezése
Allow Adding Annotations and Fields	Hivatkozások és mezők hozzáadásának engedélyezése



5. ábra Ilyen szépen kéri az Adobe Reader a jelszót

7. táblázat *A Color/Szín almenü tartalma*

General	Általános
Output Intended For (Screen/Web, Printer, Grayscale)	Milyen célra készül (Képernyő/Internet, Nyomtatás, Szürkeárnyalatos)
Use Custom Rendering Settings	Egyedi leképzési jellemzők használata
Rendering Settings (Black, Cyan, Magenta, Yellow)	Leképzési jellemzők (Fekete, Cián, Magenta, Sárga)
Frequency (10 - 1000)	Rácssűrűség
Angle (-180 - 180)	Rácselforgatás
Spot Function (Simple Dot, Line, Round, Ellipse)	Rácspontalak (Egyszerű Pont, Vonal, Kerekített, Elliptikus)

- ***Resolution for EPS Graphics***: szöveges és vektoros grafikák képfelbontása. A 300 dpi-s alapérték alá ne menjünk.

- ***Compress Text and Vector Graphics***: a dokumentumban tárolt szöveges és vektoros grafikák tömörítésének engedélyezése. Csökkenti a fájl méretét.

- ***Image Settings***: a képek/fotók tömörítése és felbontása szintén befolyásolja a fájl méretét, de a minőségére is nagy hatással van. Ezért a legjobb minőséget a tömörítés kikapcsolásával érhetjük el, valamint a ***Quality – Maximum*** értékével.

- ***Resample Images to***: egy szabadon választható felbontási értékre konvertálja a képeket.

A Security/Biztonság almenü

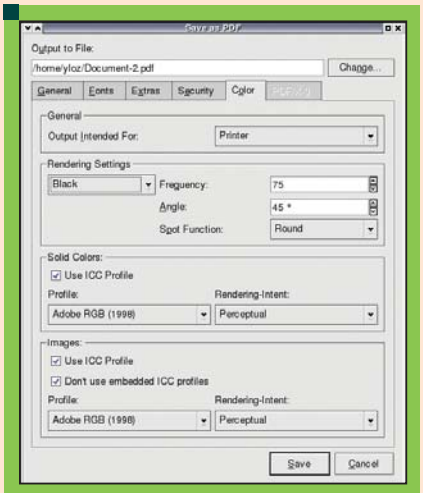
Egy hasznos lehetőség a dokumentumok illetéktelen szemek elöli elrejtésére, valamint a hozzáféréssel rendelkezők jogainak kordában tartására (*5. ábra*). A *6. táblázaton* ezen jogterületek felsorolása is megtalálható. A titkosítás ***PDF 1.3*** esetében 40 bites, a ***PDF 1.4*** esetében pedig 128 bites kódolással történik.

A Color/Szín almenü

Ha még nem volt elég a színek állítgatásából, akkor itt tovább folytathatjuk (*7. táblázat*).

- ***Output Intended For***: a célterületnek megfelelően lehet kiválasztani a kívánt színprofilt. ***Képernyőre/Internetre és Szürkeárnyalatos*** cél esetében semmit nem kell (nem is engedi) a továbbiakban állítani. Ellentétben a ***Nyomtatásra készülő dokumentum*** kiválasztásánál.

- ***Use Custom Rendering Settings***: csak a ***Nyomtatásra készülő dokumentum*** készítésénél jelenik meg. Szándékosan nincs alapértelmezettre állítva, mert nem megfelelő használat esetén nagy esély mutatkozik arra, hogy a nyomat „elszíneződik”.



6. ábra A Color/Szín almenü teljes pompájában

- ***Rendering Setting***: a felsorolásban szereplő színek rácsjellemzőit állíthatjuk át. Ez már komoly szakértelmet igényel, ezért érdemes ezeket nem állítgatni, de ha mégis szükségét érezzük, akkor nagy körültekintéssel járjunk el.

Egyszerű PDF fájl készítése

További hozzávalók: egy korábban vagy frissen szerkesztett dokumentum, háttérben duruzsoló zene és egy kényelmes ülőalkalmatosság. Ha mindez meg van, akkor el is kezdhetjük a *PDF*-fé alakítást. Első lépésként a ***Preflight Verifierrel*** (rakétát ábrázoló ikon, de megtaláljuk a ***Tools*** menüben is) keressük meg és javítsuk ki az esetleges hibákat. A ***File*** menü ***Export/Save as PDF*** almenüjét választva vagy az ***Adobe*** logóját ábrázoló ikonra kattintva csalogathatjuk elő a nem is olyan régen ismertetett ***PDF Export*** menü bővített változatát. Ami eltérés mutatkozik a két menü tartalma között az a választható ***PDF/X-3*** almenü (*8. táblázat*), valamint a kibővült ***Color*** (*9. táblázat*), és két újabb, a ***Fonts*** (*10. táblázat*) és az ***Extras*** (*11. táblázat*) almenük.

A PDF/X-3 almenü

A ***PDF/X-3***-nál külön állíthatjuk a színprofilt a bővített színtér kezelés következtében.

- ***Info String***: megadása kötelező jellegű a ***PDF*** szabvány betartásának céljából. Elegendő a készített dokumentum címét beírni.

8. táblázat *A PDF/X-3 almenü*

PDF/X-3 Output Intent	A PDF/X-3 célegységre irányuló beállításai
Output Profile	Célegység (nyomtatás) profilja
Info String	Tájékoztató
Trim Box	Szegély doboz (bal, jobb, felső és alsó oldalélek)

9. táblázat *A Color/Szín almenü eltérései*

Solid Colors	Homogén színek
Use ICC Profile	ICC profil használata
Profile	A használni kívánt profil
Rendering-Intent	A leképzés módja
Images	Képek
Use ICC Profile	ICC profil használata
Don't use embedded ICC profiles	Nem beágyazott ICC profilok használata
Profile	A használni kívánt profil
Rendering-Intent	A leképzés módja

10. táblázat *A Fonts/Betűkészletek almenü*

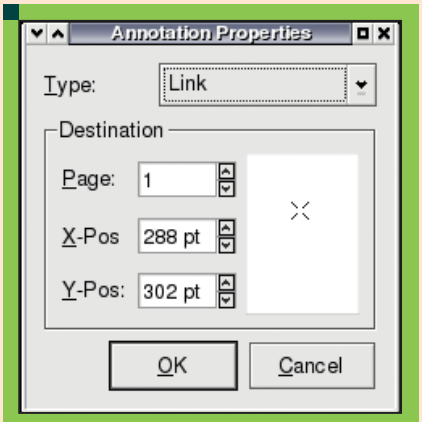
Embed all Fonts	Az összes felhasznált betűkészlet beágyazása
Subset all Fonts	Az összes felhasznált betűkészlet leírását ágyazza be
Embedding	Beágyazás
Available Fonts	Rendelkezésre álló betűkészletek
Fonts to embed	Beágyazásra kerülő betűkészletek
Fonts to subset	Beágyazásra kerülő betűkészlet leírások

11. táblázat *Az Extras/Különleges bővítmények almenü*

Enable Presentation Effects	Prezentáció Effektusok Engedélyezése
Effects	Effektusok
Display Duration	Megjelenítés Időtartama
Effect Duration	Effektus Időtartama
Effect Type (Blinds, Box, Dissolve, Glitter, Split, Wipe)	Effektus Típusa (Redőny, Doboz, Eltűnés, Csillogás, Felosztás, Törlés)
Moving Lines (Horizontal, Vertical)	Mozgó Vonalak (Vízszintes, Függőleges)
From the (Inside, Outside)	Honnan (Belül, Kívül)
Direction (Left to Right, Top to Bottom, Bottom to Top, Right to Left, Top-Left to Bottom-Right)	Irány (Balról Jobbra, Fentről Le, Lentről Fel, Jobbról Balra, Bal felső sarokból a Jobb alsó sarokba)
Apply Effect on all Pages	Effektus engedélyezése az összes oldalon
Show Page Previews	Gyorsnézet Mutatása

12. táblázat *Az Annotation Properties/Jegyzet Tulajdonságok almenü*

Text	Szöveg (emlékeztető cetli)
Link	Hivatkozás
External Link	Külső Hivatkozás
External Web-Link	Hálózati Hivatkozás



7. ábra Ahol a kereszt, ott a cél

- ***Trim box***: a fizikai lap (amire készül a nyomat) oldalaitól való eltartás mértéke.

A Color/Szín almenü eltérései

Rengeteg lehetőség kínálkozik, a színprofilok variálására (*6. ábra*). Külön a homogén színekre és külön a beillesztett képek, fotók, rajzok színeire. Itt megint csak azt tudom leírni, hogy a nyomtatást végző gépnek megfelelő profilt válasszuk, már ha rendelkezünk a szükséges adatokkal és profilokkal.

A Fonts/Betűkészletek almenü

A betűkészletek beágyazását mindenképpen engedélyezzük, mert így a fájl magával szállítja a szükséges készleteket és nem kell gondoskodni a pótlásukról.

- ***Subset all Fonts***: ezt ne használjuk. Igaz, hogy kisebb fájlméretet eredményez, viszont csak a készlet leírásokat tartalmazza.

- Az ***Embedding*** részen a felhasznált készleteket láthatjuk és a nyilak segítségével rendezhetjük, hogy melyik legyen teljesen beágyazva vagy csak a leírása kerüljön beépítésre.

Az Extras/Különleges bővítmények almenü

A prezentáció effektusok engedélyezésével apró effektusokat építhetünk be a ***PDF*** fájlunkba. Nincs akkora választék, mint egy ***OpenOffice.org***-ban, de segítségükkel látványosabbá tehetjük a dokumentációt. Minden oldalhoz külön beállíthatunk egy nekünk tetsző effektust, de akár mindegyikre állíthatjuk ugyanazt is (***Apply Effect on all Pages***).
Hogy legyen végre egy kis sikerélményünk, mentsük el a dokumentumot egy nekünk tetsző ***PDF*** típusban, és tekintsük meg művünket az ***Adobe Readerrel***. Ha semmi extrát nem tettünk bele, akkor egy egyszerű dokumentumot kaptunk, amit nem tudunk módosítani (ez egy igazán kicsi sikerélmény). Ezért mindig ügyeljünk arra, hogy az eredeti nyers dokumentum (***Scribus*** saját formátuma) még véletlenül se keveredjen el!
Az egérrel előcsalható menüben található egy ***PDF Options*** almenü (megtalálható az eszköztáron is ABC és két lábnyom ikon formájában), ez tartalmaz alapesetben két további almenüt. Ez a két almenü csak szövegdoboz és táblázat esetén használható.

- ***Is PDF Bookmark***: a kijelölt objektumból könyvjelzőt készít. Ilyen könyvjelzőket bárhol elhelyezhetünk a dokumentumban, mentéskor ne felejtsük el bejelölni az ***Include Bookmarks*** opciót.

- ***Is PDF Annotation***: szintén a kijelölt objektumot alakítja át egy vezérlő objektummá. Használatakor egy újabb almenü jelenik meg a ***PDF Options***-ban, az ***Annotation Properties***. Itt további négy lehetőség közül választhatunk (*12. táblázat*).

- ***Text***: a szövegdobozba írt szöveg a ***PDF*** dokumentumban egy kis papírlap ikon formájában jelenik meg, afféle emlékeztető cetli.

- ***Link***: az objektumra kattintva a kereső ablakban bejelölt oldalra/pozícióra teszi át a nézetet (*7. ábra*).

- ***External Link***: egy külső ***PDF*** fájl valahányadik oldalára hivatkozhatunk, szintén a kereső ablak segítségével pontosíthatjuk a hivatkozás helyét.

- ***External Web-Link***: az internetes honlapok címét érdemes így megadni, azonnal böngészhető, ha van kapcsolat.

Hivatkozások készítésénél a legcélszerűbb üres szövegdobozok alkalmazása. A betördelt szövegben, a hivatkozó szót színezzük át egy tetszőleges színre és e szó köré rajzoljuk meg az üres szövegdobozt. Így nem kell játszani a különálló objektum beillesztésével, nem töri meg a tördelést és könnyen eltávolítható, mindenféle következmény nélkül.
És nem utolsósorban így képeket is hivatkozóképessé tehetünk, vagy bármilyen más objektumot.
Az emlékeztető cetliket nyugodtan használjuk, sajnos nyomtatásban a tartalmuk nem fog látszani, de a számítógépen olvasott dokumentumban nagyon jól betölti a szövegbe ágyazott „lábjegyzet” szerepét.
És hogy a kész ***PDF*** olvashatóságát se zavarja, csak arrébb kell tenni a cetlit az útból.
E száraz anyag után a következő részben az itt ismertetett funkciók gyakorlatban történő alkalmazását mutatom be, egy rajzokkal, rétegekkel megtűzdelt PDF fájl elkészítésén keresztül.

Lelovics Zoltán
(ylozmarr@freestart.hu)
Mindig is foglalkoztatott, hogyan lehet szemmel, kézzel vagy füllel „foghatót” szépet alkotni számítógép segítségével, hogyan lehet mindezt összekötni és milyen lehetőségek kínálkoznak e cél eléréséhez.

Trust 400-V2 digitalizáló-tábla telepítése

Ki ne szeretne olyan precízen és kecsesen rajzolni számítógépen is, mint papíron tollal? Lássuk, hogyan lehet ezt megoldani Linux alatt.

Ha már próbáltunk egérrel képet retusálni, akkor tudhatjuk, hogy főleg a szűk helyeken ez nem is olyan egyszerű. Az egeret nem könnyű, sőt néha lehetetlen precízen irányítani, ami nem is olyan meglepő, hiszen marokra fogjuk.
Avagy megfordult már a fejünkben, hogy egy unalmas matekórán készített kézi rajzunkat, ami nagyon jól sikerült újrarajzoljuk számítógépen? Vagy esetleg egy kézírást felismerő programmal szeretnénk felváltani a hosszú és fárasztó gépelést? Ilyen és ehhez hasonló helyzetekre jelent jó megoldást a digitalizáló-tábla használata.

1. ábra A jószág...

Sokféle létezik, és a nagy márkák közös tulajdonsága, hogy mind képesek együttműködni a Linuxszal is. Ezeket a méregdrága típusokat azonban nem mindenki engedheti meg magának. A ***Trust 400***-as tábla azoknak jelent megoldást, akik nem tudnak vagyonokat költeni ilyen táblákra, de mégis szeretnék élvezni a digitalizáló-táblák nyújtotta előnyöket.
A ***Trust*** tábla hardverét az egyik híres cég, az ***Aiptek*** szolgáltatja. A Trust tulajdonképpen csak a burkolatot adja hozzá. Az ***Aiptek*** modulok illetve meghajtók sajnos nem kifejezetten naprakészek, ezért ***Linux*** alatt szükségünk van némi „háttérinformációra", vagyis arra, amiről ez a cikk szól.
Fontos még megjegyezni, hogy a precíz irányíthatóság mellett a ***Trust*** és más táblák is nyomásérzékenyek.
Ez egy igen hasznos tulajdonság, amit a Gimp természetesen támogat.

Szükséges összetevők beszerzése

Töltsük le a ➲ ***http://fabiana.web.elte.hu/prg/aiptek*** oldalról a legfrissebb csomagot. Ez a saját weblapom, és azért ajánlom ennek a használatát, mert az itt található csomagot kiegészítettem pár hasznos dologgal, amelyek az eredetiből hiányoznak. Aki mégis a hivatalos változat mellett dönt, az látogasson el a ➲ ***http://aiptektablet.sourceforge.net*** weblapra.
A letöltött csomagot tömörítsük ki valahova.

```
$ tar xjvf aiptek_package_20050909.tar.bz2
```

A kernelmodul telepítése

Akkor használhatunk előrefordított modult, ha a kernel már eleve tartalmazott egy régit. Ha nincs ***aiptek.ko*** modul a ***/lib/modules/2.y.z/kernel/driver/usb/input/*** alatt, akkor ugorjunk a következő fejezetre. Keressünk egy megfelelő előre lefordított kernel modult a ***prebuild*** mappában. Itt megtalálhatóak a 32 illetve a 64 bites változatok egyaránt.

```
$ cd ./prebuild/kernel_modul
```

Ha megtaláltuk, akkor másoljuk a következő helyre, de előtte szerezzünk rendszergazdai jogokat.

```
$ su
$ password: jelszó

$ cp ./2.x.y.z_arch/aiptek.ko
➥/lib/modules/2.x.y.z/kernel/driver/usb/input
```

Most teszteljük le, hogy valóban működik-e. Csatlakoztassuk a digitalizáló táblánkat és indítsuk újra a gépet. Miután újraindult töltsük be a modult root felhasználóként.

```
$ modprobe aiptek

$ lsmod
Module                  Size  Used by
snd_seq_midi            7936  0
snd_pcm_oss            50784  0
.
.
.
ieee1394               94456  2 eth1394,
                              ➥ohci1394
nvidia               4381572  24
shpchp                 91976  0
aiptek                 22016  0
realtime               10896  0
```

Ha megtaláltuk a modulunkat a kimenetben és nem kaptunk hibaüzenetet, akkor bizakodhatunk, hogy jól fog működni, de hogy biztosak legyünk még meg kell néznünk, hogy valóban ad jelet a táblánk. Ezt úgy tudjuk megnézni, hogy a megfelelő eszközbemenetet (device input) íratjuk ki. Ezzel tulajdonképpen a táblából jövő jeleket íratjuk ki nyersen a képernyőre.

```
$ cat /dev/input/eventX
```

Ahol az X valamilyen rendelkezésre álló számot takar. Jegyezzük meg jól, hogy milyen számon értük el a táblánkat. Ha ezután megpróbálunk írni a táblán, akkor mindenféle ákom-bákom karaktereket kell, hogy kapjuk kimenetként. Ekkor jól működik a kernel modulunk és nagy valószínűséggel nem kell új kernelt fordítanunk, kihagyhatjuk a következő fejezetet. Ha valahol hibát kapunk, vagy nem kapunk semmilyen kimenetet a fenti parancsra és minden rendelkezésre álló számot behelyettesítettük x helyére, akkor nem jó a modul, és új kernelt kell fordítani, ezt írja le a következő fejezet.

A kernelmodul telepítése forrásból

Mivel minden ***Linux*** disztribúció más-más módon támogatja az új kernel fordítását, ezért nem tudok kitérni minden egyes részletre, inkább a filozófiáját szeretném átadni és 2 darab konkrét példával megerősíteni. Az egyik ***Gentoo Linuxra*** lesz, a másik példa pedig ***Debian GNU/Linuxra***. Az általános kernel leírás megtalálható a ***http://www.szabilinux.hu/kernel/Kernel-HOGYAN.html*** helyen. Telepítsük a kernel forrást a disztribúciónak megfelelően cdrom-ról, vagy az internetről. Ha a fent említett disztribúció valamelyikét használjuk, akkor root-ként adjuk ki a következő parancsok valamelyikét.
Gentoo Linux alatt:

```
$ emerge -v gentoo-source
```

Debian GNU/Linuxra alatt:

```
$ apt-get install kernel-source
```

Ezzel a paranccsal még nem telepítettük a kernel forrást, csak egy listát kapunk a rendelkezésre álló verziókról. Ha telepíteni szeretnénk, akkor a fenti parancsba a listából kiválasztott konkrét verzióval együtt írjuk be. Ezután másoljuk át a csomagban lévő kernel modul forráskódját a letöltött kernel forráskódjába.

```
$ cp  aiptek_package_20050909/src/
➥linux_kernel/2.x/aiptek.c /usr/src/
➥linux-x.y.z/drivers/usb/input/
```

Ezzel a parancsa egyben lecseréltük a régi ***aiptek.c*** forrás fájlt, ami hibás működést eredményezett volna. Állítsunk be néhány szükséges dolgot a kernelünkön a következő módon:

```
$ cd /usr/src/linux-x.y.z
$ make menuconfig
```

Kapcsoljuk be a következő támogatásokat:

```
Device drivers ->
 USB Support ->
  "Support for Host-side USB" [y]
  "USB Human Interface Device (full HID) support" [y]
  "HID input layer support" [y]
  "Aiptek 6000U/8000U tablet support" [M]

Device drivers ->
 Input device support ->   "Event Interface" [y]
```

Az ***Aiptek*** támogatást modulként kell aktiválni, amit az ***m*** billentyű lenyomásával vagy a space billentyű többszöri lenyomásával lehet megtenni. A többi támogatás pedig a kernelbe fordítódik, ezeknél a 'y' karaktert használjuk. Most lépjünk ki a kernel konfiguráló programból és fordítsuk le a kernelt.
Gentoo Linux esetén (részletek a ➲ http://gentoo-wiki.com/HOWTO_Compile_a_Kernel_Manually helyen)

```
$ make bzImage && make modules_install
$ mount /boot
$ cp arch/i386/boot/bzImage /boot/vmlinuz-.2.y.z
➥# 64 bites rendszernél nem i386 van,
➥hanem x86_64.
$ cp System.map /boot/System.map-2.y.z
$ nano /boot/grub/grub.conf
```

Az utolsó paranccsal tudjuk a rendszerbetöltőt konfigurálni. Ebben nem kell mást tennünk, mint egy új bejegyzést készíteni a meglévő alapján az új kernelünkre. ***Debian GNU/Linux*** esetén (részletek a ➲ ***http://www.projektfarm.com/en/support/howto/debian_kernel_compile.html*** helyen):

```
$ make dep
$ make-kpkg clean
$ fakeroot make-kpkg --revision=custom.1.0
➥kernel_image
$ cp System.map /boot/System.map-2.y.z
$ cd ../
$ dpkg -i kernel-image-2.y.z_custom.1.0_i386.deb
```

Ez után indítsuk újra a gépet és ellenőrizzük, hogy jó kernel töltődött-e be. Mielőtt újraindítanánk a rendszert, ellenőrizzük újból, hogy mindent jól csináltunk:

- A lefordított kernelt átmásoltuk a ***/boot*** könyvtárba úgy, hogy a régit nem töröltük.
- A ***System.map*** fájlt átmásoltuk a ***/boot*** könyvtárba úgy, hogy a régit nem töröltük.
- Az új kernel számára készítettünk egy új bejegyzést a rendszerbetöltőben. A ***Debian*** esetén ez magától elkészül, de ennek ellenére ellenőrizzük!

Ha mindent rendben találtunk, akkor bármikor visszatérhetünk az eredeti kernelhez, így nem kockáztatjuk a saját rendszerünk épségét. Az újraindításnál figyeljünk arra, hogy az új kernelt válasszuk ki és ne a régit.

```
$ shutdown -h now
$ uname -a
Linux black 2.6.12.5 #4 SMP Sun Aug 21 20:55:43
```

1. lista

```
#!/bin/sh
#
# configure & setup aiptek usb tablets
#
# AUTHOR: Joost Diepenmaat - joost@zeekat.nl

. /etc/hotplug/hotplug.functions

if [[ "$ACTION" != "add" ]]; then
    exit 0;
fi

AIPDIR=`echo $DEVPATH|grep -o '[^\/]*$'`

if [[ "$AIPDIR" ==  "" ]]; then
    mesg "Can't find aiptek tablet /sys entry"
    exit 1;
fi

AIPDIR="/sys/bus/usb/drivers/aiptek/$AIPDIR"
INPUT_PATH=`cat $AIPDIR/input_path`

if [[ ("$INPUT_PATH" == "") || ("$INPUT_PATH" ==
➥"/dev/input/") ]]; then
    mesg "input_path is empty: try replugging the
    ➥tablet"
    exit 0
fi

mesg "Link létrehozása /dev/input/aiptektablet to
➥$INPUT_PATH"

if [[ -L /dev/input/aiptektablet ]]; then
    rm -f /dev/input/aiptektablet
fi
ln -s $INPUT_PATH /dev/input/aiptektablet


mesg "A tábla konfigurálása"
# A gombok korrekciója
echo "middle" >$AIPDIR/mouse_right
echo "right" >$AIPDIR/mouse_middle
echo "execute" >$AIPDIR/execute
```

```
➥CEST 2005 x86_64 AMD Athlon(tm) 64 Processor
➥3500+      AuthenticAMD GNU/Linux
```

A kimenet persze mindenkinél más és más. Végezzük el az előző fejezetben leírt tesztelési műveletet a frissen fordított kernelünkkel is. Ha sikerrel jártunk tegyük automatikussá az aiptek modul betöltését a következő módon.

```
$ echo "aiptek" >> /etc/modules.autoload.d/
➥kernel-2.6 # 2.6.z kernel esetén az utolsó szám
➥6-os (lásd fent).
```

Ezzel végeztünk a kernel oldali telepítéssel, ehhez már nem kell hozzányúlnunk.

Hotplug rendszer használata

Előfordulhat, hogy a fent meghatározott eszköz elérési út (*/dev/input/eventx*) megváltozik, az x helyén más szám lesz

2. ábra Működik!

mint addig. Ilyen eset az, amikor új hasonló eszközt szeretnénk telepíteni. Hogy ez elkerülhető legyen, minden indulásnál egy szkript ellenőrzi, hogy melyik eszközön érhető el a táblánk, és arra az elérési útra készít nekünk egy szimbolikus láncot. Ezenkívül a szkript elvégez a nyomógombok cserét, hogy kézre álljon minden. Ha nem rendelkezünk hotplug-al, akkor telepítsük.

Gentoo Linux esetén:

```
$ emerge hotplug
$ rc-update add hotplug boot
```

Debian GNU/Linux esetén:

```
$ apt-get install hotplug
```

A szkriptet az *1. lista* mutatja.
Másoljuk be a szkriptet a megfelelő helyre:

```
$ cp aiptek_package 20050909/hotplug_script/
➥aiptek /etc/hotplug/usb/
```

Ezután másoljuk a következő fájlt ugyanide, de előtte győződjünk meg arról, hogy a tábla csatlakoztatva van.

```
$ cp /lib/modules/*kernel-version*/modules.usbmap
➥/etc/hotplug/usb/aiptek.usermap
```

Most már biztosak lehetünk abban, hogy a jó eszközt használjuk majd a grafikus szerver konfigurációs fájljának beállításakor.

A grafikusszerver-modul telepítése előre fordított modulból

Hasonlóan a kernel modulhoz, itt is az előre lefordított modul telepítésével és tesztelésével kezdjük, csak most az *X.org* vagy *Xfree86* grafikusszerver-modult szeretnénk telepíteni.

```
$ cd aiptek_package 20050909/prebuilt/x11_modul/
```

Ha kiválasztottuk a megfelelő mappában lévő modult, akkor másoljuk a következő helyre.

```
$ cp ./2.x_arch/aiptek_drv.o
➥/usr/X11R6/lib/modules/input/aiptek_drv.o
```

Ha ezzel megvoltunk tesztelhetjük a modult. Ehhez ugorjuk át a következő fejezetet és kezdjük meg a tábla beállítását.

A grafikus szerver modul telepítése forrásból

Csak akkor kell a grafikus szerverhez tartozó ***aiptek*** modult forrásból újrafordítani, ha 2.4-es kernelünk van és 64 bites rendszeren futtatjuk, mivel csak az ilyen konfigurációhoz nincsen megfelelő előre fordított modul (meghajtó). Szükségünk lesz a jelenlegi grafikus szerver forráskódjára. Ezt disztribúciótól függően tudjuk megszerezni, illetve telepíteni CD-ROM-ról vagy az internetről. Az eredeti forráskódokat cseréljük le a csomagban találhatóakra.

```
$ cp aiptek_package 20050909/src/
➥xserver_input_driver/* \
<x11 szerver name>/work/xc/programs/Xserver/
➥hw/xfree86/input/aiptek/
```

Ezután fordítsuk le az ***X*** szervert, ***de ne telepítsük***! Az újonnan lefordított ***aiptek_drv.o*** modult másoljuk át a már telepített és használt ***X*** szerverünk moduljai közé.

```
$ cp <x11 szerver name>/work/xc/programs/Xserver/
➥hw/xfree86/input/aiptek/aiptek_drv.o \
/usr/X11R6/lib/modules/input/aiptek_drv.o
```

Ezzel lecseréltük az eredeti, de rossz ***aiptek_drv.o*** modult az újonnan fordított és jól működő modulra. Mehetünk beállítani a táblánkat.

A tábla beállítása

Nyissuk meg a grafikus szerverünk konfigurációs fájlját a kedvenc szövegszerkesztőnkkel root felhasználóként.

```
$ gedit /etc/X11/xorg.conf
```

Ezután adjuk hozzá a következő sorokat:

```
Section "InputDevice"
 Identifier "pen"
 Driver     "aiptek"
 Option     "Device"    "/dev/input/aiptektablet"
 Option     "Type"      "stylus"
 Option     "Mode"      "absolute"
 Option     "Cursor"    "stylus"
 Option     "USB"       "on"
 Option     "KeepShape" "on"
EndSection

Section "InputDevice"
 Identifier "cursor"
 Driver     "aiptek"
 Option     "Device"    "/dev/input/aiptektablet"
 Option     "Type"      "cursor"
 Option     "Mode"      "absolute"
 Option     "Cursor"    "puck"
 Option     "USB"       "on"
 Option     "KeepShape" "on"
EndSection

Section "InputDevice"
 Identifier "eraser"
 Driver     "aiptek"
 Option     "Device"    "/dev/input/aiptektablet"
 Option     "Type"      "eraser"
 Option     "Mode"      "absolute"
 Option     "Cursor"    "stylus"
 Option     "USB"       "on"
 Option     "KeepShape" "on"
EndSection
```

Ezután adjuk még hozzá a három új `InputDevice` részt a `ServerLayout` szekcióhoz.

```
Section "ServerLayout"
 Identifier   "Default Layout"
 Screen       "Default Screen"
 InputDevice  "Generic Keyboard"
 InputDevice  "IntelliMouse Explorer"
              ➥"CorePointer"
 InputDevice  "pen"       "AlwaysCore"
 InputDevice  "cursor"    "AlwaysCore"
 InputDevice  "eraser"    "AlwaysCore"
EndSection
```

Mentsük el a fájlt! Ezzel beállítottuk a digitalizáló táblánkat és minden funkciója működőképes. De persze csak miután újraindítottuk a gépünket (elég csak az ***X***-et újraindítani).

Tesztelés Gimppel

Miután újraindítottuk a gépünket, már használhatjuk is a táblánkat, mint kiegészítő egér. A nyomásérzékenységet viszont csak rajzolás közben élvezhetjük. Ehhez indítsuk el a ***Gimpet*** és a ***Fájl -> Beállítások -> Bemeneti eszközök -> Kiterjesztett bemeneti eszközök beállítása -> Pen ->*** helyen levő értéket állítsuk át ***Képernyő*** módra. Most indítsunk el egy valamilyen méretű rajzlapot és firkáljuk rá különböző nyomáserősséggel. Nagyszerű, működik a digitalizáló-táblánk. További jó rajzolgatást!

Fábián Attila (e-mail: fabiana@elte.hu)
Az Eötvös Loránd Tudományegyetem Természettudományi Karán vagyok vegyészhallgató. 4 éve használok Linuxot. Ha ezek után még mindig marad szabadidőm, akkor 3D grafikával foglalkozom vagy biciklizek.

KAPCSOLÓDÓ CÍMEK

Az Aiptek linux-os oldala:
➲ http://aiptektablet.sourceforge.net/

Az Aiptek meghajtó bővített kiadása:
➲ http://fabiana.web.elte.hu/prg/aiptek

Fraktálok – avagy a matek akár szép is lehet

Most egy olyan témáról szóló cikk következik, ami biztosan nem jött volna létre, ha nincsenek számítógépek. Olyan mennyiségű számítást igényel ugyanis a cikkben szereplő képek előállítása, hogy az számítógép nélkül elképzelhetetlen.

A fraktálgeometria tudománya alig három évtizedes, tehát fiatalnak tekinthető tudományág. A természetben előforduló, más matematikai módszerekkel elég pontosan nem meghatározható, önhasonló objektumok leírásával, tanulmányozásával foglalkozik. A fraktál kifejezést először Benoít Mandelbrot lengyel származású matematikus használta, 1975-ben. A szó a latin fractus szóból ered, melynek jelentése törött, töredezett. A természetben számtalan olyan komplex objektum található melyek leírására jól használhatóak a fraktálok, mint például a felhők, hegyek, fák, szigetek partvonalai, ill. az egyik legszemléletesebb a páfrány. Ha alaposan megnézzük a páfrány levelét, akkor láthatjuk, hogy a kisebb részek formája erősen hasonlít az egész levél alakjára.
Manapság a faliképektől kezdve, a képernyő pihentetőn keresztül egészen a csomagolópapírokig nyúlik a fraktálképek felhasználásának hosszú sora. Aki számítógépes grafikai programot használ, az gyakran találkozhat fraktál alapú textúrákkal is.
A fraktálokat leíró képletek komplex számokat tartalmaznak, melyek egy valós és egy képzetes részből (például $z = 3 + 2i$) állnak. A komplex számokkal végzett műveletek szabályai némileg eltérnek a megszokottól, például a képzetes rész négyzete -1. A további szabályok megtalálhatóak például a *Négyjegyű függvénytáblázatok* című közismert műben.
A fraktálképek kiszámítása egy iteratív, azaz ismétlődő folyamat. Az ismétlések száma meghatározza a fraktálkép minőségét, és a számítás idejét.
A számítás során gyakran használnak rekurzív eljárásokat. A rekurzív eljárás önmagát meghívva ismétli meg a szükséges lépéseket.
A fraktálképet a következő paraméterek határozzák meg döntően:

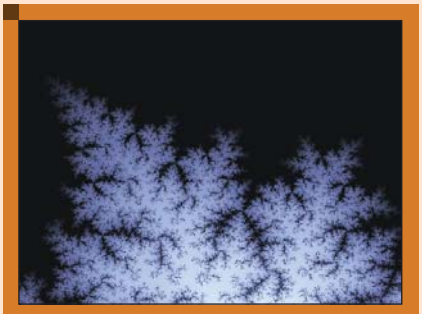
- kiinduló egyenlet
- iterációk száma
- nagyítás
- a megjelenítéshez használt színek, azaz a paletta

A legismertebb fraktálok a ***Mandelbrot*** ($f(z)=z^2+c$, ahol c konstans) és a ***Julia*** halmazok.
Ennyit röviden a fraktálok matematikájáról általánosságban, a részletesebb tárgyalását meghagyom a matematikusoknak. Amúgy ha valaki többet szeretne tudni a témáról az interneten is bőséggel talál információt.
Én a továbbiakban két fraktálok képeit előállító program fogok bemutatni.

FractalView

A program futtatásához szükség van minimum 1.4-es java futtató környezetre (én a Blackdown 1.4.2-t használtam). Lépjünk be abba a könyvtárba, amelyikbe a programot kicsomagoltuk és a **java FractalView** parancs kiadása után kezdhetjük a fraktálok felfedezését.
Egy mai 2 GHz-es processzor esetén kb. 3 másodperc múlva jelenik meg egy kép, amennyiben az alábbi billentyűkkel megváltoztattuk a fraktálképhez tartozó valamelyik paramétert. A számítási folyamatot a képernyő alján megjelenő vonal is jelzi.

Fractal View 1.03
Méret: 200 KB
Weblap: ➲ *fractalview.sourceforge.net*
Licensz: GPL
Igényelt grafikus könyvtár: ---
Támogatott operációs rendszerek: Linux, BSD, Windows, stb.



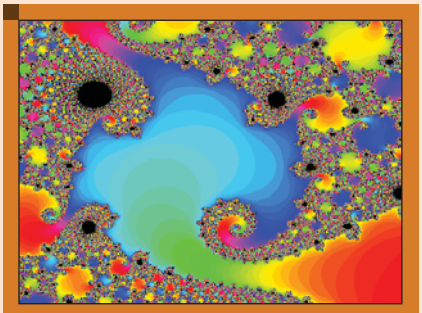
1. ábra A Mandelbrot halmaz egy kis részlete

2. ábra Mint egy repedés a falon

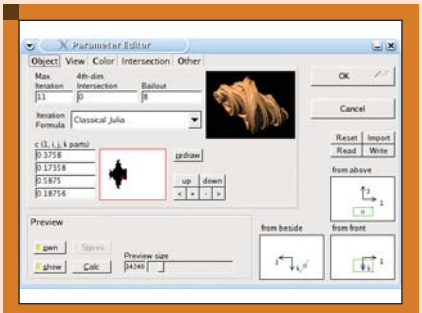
A lementett képek a program könyvtárának *FractalImages* alkönyvtárába kerülnek. Ha a program használata közben átváltunk más programra, eltűnhet a kiszámolt kép, ezt a szóköz billentyű megnyomásával helyrehozhatjuk.

1. táblázat A Fractalview kezelése során használható fontosabb billentyűk

F1 vagy h	Billentyűparancsok megjelenítése / eltüntetése
Szóköz	Újraszámolja a képet az aktuális beállításokkal
A,a,S,s	200, 100, 20, 10 db PNG képet exportál animáció készítéshez, miközben a képernyő közepén lévő részt nagyítja
n,m	Váltás a különböző fraktál típusok között
c	A kép egy pontjára kattintva választhatunk konstans értéket az egyenlethez
r	Véletlenszerű konstans az egyenlethez
[ ]	Váltás az előre beállított konstansok között
:	Visszaállítja a nézőpontot a fraktálhoz tartozó alapértelmezésre
.	Az aktuális nagyítás mellett a fraktál közepét mutatja meg
p A	megjelenő paletta egy pontjára rábökve választhatunk színt
0 – 9	Palettaváltás
q vagy ESC	Kilépés
Egér bal gomb + húzás	Adott területe körbe kerítve nagyíthatunk rá
Egér jobb gomb	A kép mentése
Egér középső gomb vagy ;	Visszaállítja az alapértelmezett nézőpontot
Egér görgő	Nagyítás / kicsinyítés



■ *3. ábra* Azoknak, akik szeretik, ha valami „jó színes"



■ *4. ábra* Itt állíthatjuk be a paramétereket

Sajnos néhány esetben megmerevedik a program, és csak a program újraindítása után folytathatjuk a fraktál képekben való gyönyörködést.
Ha van a gépen ***MPlayer***, akkor az animációhoz lementett képeket a következő paranccsal alakíthatjuk tömörítetlen mozgóképpé.

```
mencoder mf://movie*.png -mf
➥w=256:h=192:fps=25:type=png
➥-ovc raw -oac copy -o
➥output.avi
```

Mivel a program ***Java*** alapú, így más java képes operációs rendszerek alatt is gond nélkül használható. A program előnye a platform függetlenség, és az egyszerűség. Elsősorban a fraktálokkal csak most ismerkedőknek ajánlanám.

Quat

A program készítője, ezt írja: *„A célom az volt, hogy 3 dimenzióban jelenítsem meg a fraktálokat. Az eredmény pedig a térben szabadon lebegő, csavart, rögök lettek, melyek olyanok, mint a nyers tészta."* (Ha valaki látott, vagy gyúrt már kenyeret, vagy kalácstésztát, akkor tudja mire gondolt.)

Quat 1.20
Méret: 500 KB – 1MB
Weblap: ➲*www.physcip.uni-stuttgart.de/phy11733/quat_e.html*
Licensz: GPL2
Igényelt grafikus könyvtár: ---
Támogatott operációs rendszerek: Linux, DOS, Windows

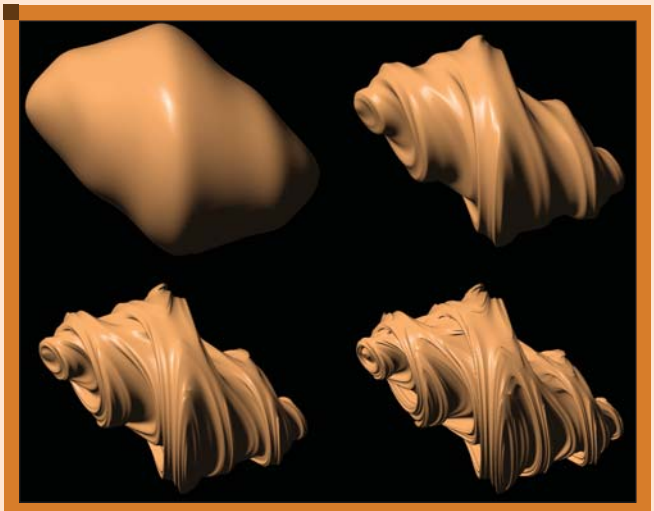
A fraktálok komplex számokra épülő képlete teljesen megegyezik a 2 dimenziós programokban használttal. A 3 dimenziós megjelenítéshez az úgynevezett „Hamiltonian quaternions"-t használja fel. A „quaternions" a komplex számokat négy részből állónak tekinti (1 valós és 3 képzetes) ellentétben a hagyományos 1 + 1 -es felbontással. Ez az eljárás alkalmas lenne 4 dimenziós fraktálok kiszámítására is, de a program csak a 3 dimenziósat számolja ki (a 4. érték konstans). A készítője viszont a dokumentációban megígéri, hogyha valaki kifejleszt egy 4 dimenziós megjelenítőt, akkor ő rögvest átalakítja a programját.

Parameters menü

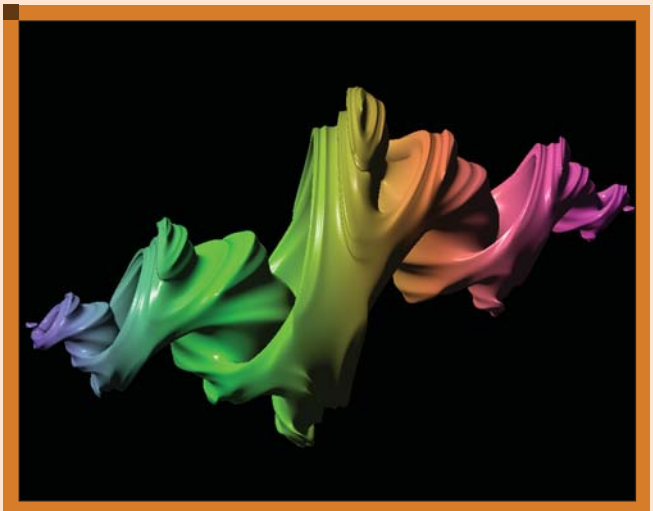
Ebben a menüben adhatjuk meg, tölthetjük be fájlból, vagy menthetjük el a számításhoz szükséges paramétereket. A ***Max Iteration*** – a kép részletességét határozza meg, az ***Iteration Formula*** – az alakzatot, az ***i,j,k paraméterek*** pedig a nézőpontot. A nézőpontot konkrét számmal is megadhatjuk, beírva az i,j,k-nak megfelelő mezőbe, vagy az egérrel a jobb alsó sarokban található téglalapokban állíthatjuk be. A beállító ablak ***Intersection*** fülén adhatunk meg metszősíkot, amivel elvágjuk az adott formát, és így még érdekesebb képet kaphatunk. Az ***Other*** fülnél beállíthatóak a teljes kiszámítandó kép méretei, világítási és élsimítási paraméterei. A ***Calc*** gombra kattintva kaphatunk előnézeti képet, a beállított értékek alapján.

Calculation menü

Mikor beállítottuk a paramétereket, akkor innen indíthatjuk vagy a teljes számítási folyamatot, vagy a ***ZBuffer*** kiszámítását. Ha a ***Zbuffert*** választjuk, akkor a program csak a fraktál formáját számolja ki, és utána már csak a színeket és a fényeket kell kiszámolnia a következő lépésben, ami már sokkal gyorsabban megy.



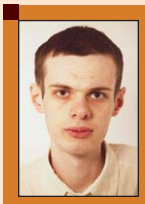
5. ábra Az iterációk számának növelésével elért eredmény (5-8-10-12)



6. ábra Egy szép színátmenetes kép

7. ábra Olyan színes, mint némelyik papagáj

néhány másodperc kell a kép befejezéséhez, miután kiválasztottuk a megfelelő színeket.
Ez a program a 3 dimenzió révén sokkal érdekesebb képek előállítására képes, mint az előző. Sokkal tovább lehet vele úgy kísérletezni, hogy mindig új formákat fedezzünk fel.
Ha csak a keresőbe írjuk, be például a *fractal*, vagy *mandelbrot* szavakat, akkor is több százezer képet találunk az interneten. Nagyon változatos képeket lehet létrehozni a különböző fraktálgeneráló programokkal. Remélem mindenki megtalálja, vagy megalkotja a neki tetszőt. Sok sikert, és sok szép képet kívánok!



Szabolcsi Csaba
(szabolcsi@walla.com)
A Linuxot 1994-ben ismerte meg, jelenleg Gentoo-t használ. Szeret olvasni és főzni.

Image menü

Ha tetszik a kiszámított kép, akkor itt menthetjük el. A program a *PNG* fájlban a fraktál paramétereit is tárolja. Ha nem felel meg az eredmény, akkor a *Close* menüponttal le kell zárni az adott képet, különben nem módosíthatjuk a paramétereket.

Zbuffer menü

Ez egy rövid menü, megnyitni, bezárni, és elmenteni lehet a *ZBuffert*. Ha ezt elmentettük, akkor már csak

KAPCSOLÓDÓ CÍMEK

Wikipédia a fraktálokról: ➲ http://en.wikipedia.org/wiki/Fractal

Fraktálok a Sulineten: ➲ http://www.sulinet.hu/tart/cikk/ag/0/22128/1

Művészi fraktálok, akár nyomtatásban is: ➲ http://fractorama.com

3D fraktálok: ➲ http://www.physcip.uni-stuttgart.de/phy11733/gallery.html

Fraktálok egymásra rétegezve: ➲ http://spanky.triumf.ca/www/fractint/lee/lee.html

Művészi képek: ➲ http://www.fractalus.com/gumbycat/

Egy másik galéria: ➲ http://www.faemalia.net/Fractals/

Mozgóképek: ➲ http://ericbigas.com/fractals/