

CHIP SPECIAL

## LINUX

Én és a Linux ● A Linux rövid története ● Samba –  
 kapcsolat a rendszerek között ● Linux az irodában  
 ● File-rendszerek ● Fwmm ablakkezelők  
 ● Motif, CDE, AcceleratedX ● A Debian Linux  
 és telepítése ● Programozunk X-Window alatt!

C  
D  
-  
R  
O  
M

GNU

Linu

N

U

X

# Mi a leggyorsabb? • Mi a legjobb? • Mit vásároljunk? • Mire használjuk? • Hogyan kezeljük?

A lap, amely választ ad a számítástechnika legtöbb kérdésére  
132–148 színes oldalon és CD-ROM-on



## Ha most előfizet, még olcsóbb!

A CHIP Magazin 1 példányának ára 784 forint.  
Ha október 31-ig előfizet, vagy megújítja előfizetését, Önnek egy példány – CD-ROM-mal együtt – csak 493 forintba fog kerülni!  
Érdeklődjön: Vogel Publishing Kft., 1300 Budapest, Pf. 210, telefon: 149-4768,  
fax: 252-3004, Fehér Ildikó

Czakó Krisztián

# Bevezető



Magyarországon egyedülálló kiadványt tart az Olvasó a kezében, hiszen kizárólag Linuxszal foglalkozó lap magyarul még sosem jelent meg. Reméljük, ez nem egyszeri kísérlet, hanem egy rendszeresen megjelenő kiadvány kezdeménye lesz.

A Linux operációs rendszer napjaink egyik legjobb és legsokoldalúbb rendszere, mind az irodákban, mind az Interneten. Talán nem tévedés azt állítani, hogy a világ leggyorsabb hálózati szervereként is használható, ezzel még így is képességeinek csak kis részére utaltunk, hiszen a Unix rendszerek a kis- és középgépeken is a legnagyobb biztonságot (működési stabilitást, betörésvédelmet) adó rendszerek.

Jelen kiadványunkban szeretnénk megismertetni Olvasóinkkal a Linux rendszert, annak képességeit és sokoldalúságát, illetve sok hasznos tippet adni a kezeléséhez, profiknak pedig a programozásához.

A cikkek mellett egy CD-t is tartalmaz a lap, melyen megtalálják az egyik legjobb Linux disztribúciót – a Debian 1.3.1-es változatát –, amely profiknak, komoly szerverek építéséhez ideális, biztonságát tekintve pedig a legjobb. Az, hogy a profiknak ideális rendszer, nem jelenti azt, hogy kezdők számára nem kezelhető. Nekik azzal is segítünk, hogy a telepítőprogram és a kezelési útmutató magyar fordításban került a CD-re.

Már megszokottá vált, hogy a CHIP Magazin CD-mellékletén 100 megabyte körüli Linux archívumot teszünk közzé hónapról hónapra. Eddig már két teljes RedHat disztribúció is megjelent. Ígérjük, hogy mostantól a RedHat csomagok mellett ott lesznek a Debian csomagok is, és ha a „helyszűke” engedi, a forrást is közöljük.

Akik nem ismerik a Linuxot, illetve a GNU GPL-t, jogosan kérdezhetik, hogy egy ilyen komoly színvonalú operációs rendszer miért ingyenes. Ahhoz, hogy erre a kérdésre jó választ adhassunk, rá kell világítani arra a mostanában ritkán hangoztatott tényre, hogy egy szoftver előállításá és annak kereskedelmi ára között semmiféle összefüggés nem található. Ahhoz a bevételhez képest, ami egy szoftver eladásából keletkezik, az előállítási költ-

ség elhanyagolható. A komoly kiadásokat a reklámtevékenység és a support fenntartása jelenti. A Linux és a többi szabad szoftver készítőit is megfizetik a munkájukért, kit a munkahelyén, kit az FSF (Free Software Foundation), így ők sem ingyen dolgoznak. Emellett igen sokan pusztán lelkesedésből készítenek el egy-egy magas színvonalú programot, vagy éppen azért, mert szűkségük van rá. Az így elkészült programok aztán bekerülnek a szabad szoftverek népes táborába, bárki használhatja, módosíthatja őket. Ez utóbbi tény is nagyon sokat nyom a latba, mert a szabad szoftverek forráskódjukkal együtt érhetőek el, azt bárki módosíthatja, majd ha a módosítást visszaküldi az eredeti szerzőnek, az beépíti a programba. Ez a titka a szabad szoftverek és így a Linux gyors fejlődésének, sokoldalúságának. A forráskód elérhetőségének van még egy igen érdekes hatása: aki már programozott életében, tudja, hogy egy program minősége és fejlesztetősége nagymértékben az eredeti verzió forráskódjának minőségén múlik. Ön kiadna a kezéből egy összegányolt kódot? Valószínűleg nem, s ezt ezen programok írói sem teszik. E kényszer azonban nem hat a kereskedelmi programok készítőire, hiszen a forrást senki sem láthatja...

Ahogy azt fent már írtuk, egy teljes Debian Linux 1.3.1 disztribúció került a CD-re, magyar telepítő-programmal, magyar nyelvű használati útmutatóval. Ez kicsit bővebben közel 600 megabyte-ot jelent, amiben benne van igen sok program, Web-szervertől a játékokig, szövegszerkesztőkig, segédprogramokig. A maradék kevéske helyre többek között a Java Development Kit legújabb verziója került. A JDK amellett, hogy lehetővé teszi a Java binárisok futtatását Linux alatt, teljes fejlesztői környezetet is ad. Továbbá felkerült a CD-re a Magyar Linux-lista és Linux-kezdő-lista archívuma, melynek leveleiből sok jó ötletet lehet meríteni mindenféle Linuxszal kapcsolatos – néha nem annyira kapcsolatos – témákból. A CD-n kaptak helyet még: a Linux Gazette – amely egy Interneten terjesztett ingyenes Linux-magazin – eddig megjelenő számai, és természetesen mintaféle-ok az e CHIPTárban megjelent cikkekhez, rengeteg dokumentáció és az újságból helyszűke miatt kiszorult írások is. ■

# Tartalomjegyzék

# Linux

Bevezető .....	3
Tartalom .....	4
<b>Belépő egy szabad világba</b>	
Én és a Linux .....	5
A szoftvermozgalom és a GNU .....	9
A Linux rövid története .....	12
<b>A Linux élesben</b>	
Levezés Marssal .....	16
Linux a hálózat-managementben .....	17
Dosemu, avagy DOS a Linux alatt .....	20
SaMBA – kapocs a rendszerek között .....	23
Linux az irodában .....	30
Trace-eljünk Linux alatt! .....	33
Merre, hogyan magyarul? .....	35
<b>File- és ablakrendszerek</b>	
File-rendszerek .....	41
Az X-Window System .....	43
Az Fvwm ablakkezelő .....	48
Az Fvwm2 ablakkezelő .....	53
AfterStep .....	57
Motif, CDE, AcceleratedX .....	59
<b>Linux-disztribúciók</b>	
Hol leljünk Linuxot? .....	63
A SuSE disztribúció .....	65
A Debian Linux és telepítése .....	68
<b>Programozunk!</b>	
Programozunk X-Window alatt! .....	75

## CD-nk Linuxról

A CD-nken a Debian GNU Linux 1.3.1-es disztribúciója található meg.

A CD-re került Debian telepítőprogramjának nagy részét magyarra fordítottuk (~90%-ban) és a rendszer bootolhatóvá tettük. Ezek miatt a CD-n található korábbi dokumentációkkal szemben a rendszer telepítése az alábbi módokon lehetséges:

– A megfelelő BIOS-szal és ATAPI CD-vel rendelkezők bebootolhatják a CD-t, ezzel elkezdődik a telepítés.

– Ha a CD nem bootol, akkor DOS alól – nem Windows DOS-ablaktól! – a \BODISK-13\MAGYAR\INSTALL.BAT indításával kezdetük a telepítést.

– Ha a Linux nem ismeri meg a CD-meghajtót (régembi vagy ismeretlen típus, vagy nem támogatott SCSI vezérlő „lóg”), vagy hálózaton keresztül telepítünk, akkor először a telepítő floppykat kell legyártanunk.

A telepítő floppyk image-file-jai is a \BODISK-13\MAGYAR\ könyvtárban találhatók meg. 1,2 Mbyte-os meghajtóban való bootoláshoz a RESC1200.BIN és DRV1200.BIN image-ek, 1,44-eshez a RESC1440.BIN és DRV1440.BIN image-ek szükségesek. A hálózati telepítéshez a BASE-1.BIN, BASE-2.BIN,... BASE-5.BIN készlet is szükséges.

A floppyk legyártásáról és a telepítés folytatásáról részletesen a „A Debian Linux és telepítése” c. cikkben olvashatnak részletesen.

## CHIPTÁR: Linux

Felelős kiadó: **Ivanov Péter** ● Sorozatszerkesztő: **György György** ● A kiadvány szervezője: **Czakó Krisztián** ● Szerkesztők: **Bata László** és **Nagy Zoltán** ● Olvasószerkesztő: **Nagy Anna**  
● Címábrakftika: **Szűcs** ● Kiadja és terjeszti a **Vogel Publishing Kft.** ● Megvásárolható a kiadvány: 1139 Budapest, Hajdú u. 42-44. ● Megrendelhető telefonon a (36-1) 149-4768-as vagy faxon a 252-3004-es számon. ● Hirdetéseket: **Vogel Publishing Kft.** Reklámiroda, Levelezési cím: 1300 BUDAPEST 3. PF. 210, telefon/fax: (36-1) 252-3641 ● A színborítás és a tördelés a **Vogel Publishing** Kiadóban, **QuarkPress 3.1**, **Adobe Illustrator 5.0**, **Adobe Photoshop 2.5**, **Adobe Super ATM** és az **Adobe Dimensions** programok segítségével készült.

● Monitorozás és nyomás: **Veszprémi Nyomda Rt.** Felelős vez.: **Erdős András** vezérigazgató

● Copyright © „CHIP” **Vogel Publishing Kft.**, Budapest, Magyarország ● A közölt cikkek fordítása, utányomása, sokszorosítása,

valamint adattrendszerekben való tárolása kizárólag a kiadó engedélyével történhet. A megjelentett cikkeket szabadalmi vagy más védettségre való tekintet nélkül használjuk fel.

HU ISSN 1219-4522

Nyitrai Tamás  
(defiant@tel-  
net.hu)

# Én és a Linux

**Egy hideg novemberi napon kezdődött, egy ismerősöm valami Linuxról mesélt nekem, amit most rakott fel a gépére.**

Nem különösebben izgatott a dolog, mert eléggé megszoktam már, hogy ismerőseim állandóan mindenféle programokat raknak fel a gépeikre, s azt mesélik, hogy „milyen fantasztikus a 3 dimenziós környezet, és óriási, ahogy a hangja szól”. Szóval csak beszélt, beszélt, arról regélt, hogy mi mindent tud, hogy egyszerre futtat programokat különböző ablakokban, és azt nagyon gyorsan teszi, s ráadásul gyakorlatilag ingyen van, csak a CD-t kell megvenni, de az is csak 3-4 ezer forint.

Ekkor már jobban érdekelt ez a – hogy is mondtam? – Linux, lelki szemeim előtt megjelent egy újabb windows program, amit Linuxnak hívnak, s ezt valamiért úgy sikerült megalkotniuk, hogy gyorsabban fut, mint más windowsos társai.

„És nem fagy le?” – kérdeztem. „Eddig még nem sikerült” – válaszolta ismerősöm, s közben mosolygó arccal gondolt otthoni gépére, amin ott a Linux, és további csodás dolgokat művelhet vele.

## Ismeretségünk története

Megbeszéltünk egy időpontot, amikor felmentem hozzá megtekinteni ezt a valamit, amiről annyira áradozott. Elindította gépét, betett egy lemezt, és hamarosan megjelent a „LILO:” felirat, ezt még egy rakás más felirat is követte, amiről gőzöm sem volt, hogy mit jelenthet. Amikor a feliratfolyam abbamaradt, megjelent egy rejtélyes „darkstar login:” üzenet. Bepötyögött valamit, ami után én azt vártam, hogy megjelenjen valami színes logó, vagy valami Windows-szerű munkaablak, de továbbra is csak egy villogó kurzort láttam.

Azt közben tisztáztuk, hogy a Linux futtatásához nem kell Windows, s az ismerősömnek azt is sikerült elmagyaráznia, hogy miért kell belogolni, de a beigért ablakokat továbbra is hiányoltam. „Mutass akkor valamit!” – mondtam neki kissé hitetlenkedve. Erre előbb elővárásolt egy Nortonhoz hasonló

file-kezelőt, majd nyomott valamit, amitől ez eltűnt, s újra csak a „darkstar login:” jelent meg. Na, mondom, ez lefagyott, s most léphet be újra. Be is lépett, de akkor megint megnyomott valamit, mire visszajött a file-kezelő. Ejha, ez aztán már varázslás, itt lehet kapcsolgatni a karakteres programok között.

Ismerősöm elővett egy CD-lemezt és berakta. Bepötyögött valami iszonyú hosszú és bonyolult utasítást, csak azért, hogy a CD tartalmát meg tudja nézni abban a kék nortonos valamben. Hú, ez aztán nem valami felhasználóbarát, ha ennyit kell gépelni, ráadásul magától észre sem veszi, hogy beraktam egy új CD-t – mondtam, de ismerősöm megnyugtatóan, hogy lehet írni „szkripteket” a dolgok leegyszerűsítésére, s elmagyarázta, hogy azok olyanok, mint DOS-ban a batch file-ok.

Ezek után folytatta varázslói tevékenységét. Elindeített újabb programokat, többek közt egy rejtélyes Povray nevezetűt, ami számolgatott valami képet, s közben ide-oda tudott váltogatni: hol ezt néztük, hol azt, de ablakok még mindig sehol. Erre beírt megint valami utasítást, amire megjelent egy kék felület, a képernyő sarkában valami rejtélyes négyzetekkel, de ikonok sehol. Hogy lehet innen programokat indítani, meg egyáltalán dolgozni?

Eléggé barátságatlannak tűnt a dolog, de mégse fagy le, gyors, és a programok között lehet kapcsolgatni, ez valami szenzációs. Elhatároztam, hogy én is felteszem a gépemre. Beszereztem egy CD-t, de azzal nem igazán boldogultam, így aztán ismerősöm sietett segítségemre, aki saját CD-jét hozta el. Eléraktam a klaviatúrát, s elmondtam neki, hogy azért rohagásztam pöntek este winchesterért, hogy legyen hova felrakni a Linuxot, sőt örömmel jelentettem, hogy meg is formatáltam neki.

Ekkor azonban kiderült, hogy ez nem ilyen egyszerű, mert újra kell particionálni vagy mi, és ráadásul valami fdiskkel kell búvészkedni, mert a Linuxnak saját file-rendszere van (bár akkor még egyik szó jelentését sem ismertem). Utóbbi műveleteket tapasztalt kézre bíztam – ismerősömrre, mert ő már csinált ilyet.

Miután végezett winchesterem átalakításával, készített két floppyt, amit szépen beadagolt a gépnek egymás után. Ezzel elkezdődött a hadművelet második része, amikor is egy Slackware nevezetű

program mindenféle kérdéseket tett fel, hogy akarok-e ilyen-olyan csomagokat. Mondtam neki, hogy persze, én mindent akarok. Ezután mint egy örült elkezdett felpakolni mindent a CD-ről. Megkérdeztem erre ismerősömet, hogy mi tart ennyi ideig, mire ő közölte, hogy le kell fordítania minden programot a forráskódjából. Én csak elhülve ültem a képernyő előtt, s vártam, hogy mi sül ki az egész-ből.

A telepítés befejeződött. Újraindítottuk a gépet, s nálam is megjelent a „darkstar login:” nevezetű kiírás. Ekkor ismerősöm megmutogatta, amit még addig nem: ha nem tudom egy utasítás vagy file nevét, akkor meg kell nyomnom a [TAB]-ot, s a Linux kiegészíti nekem, vagy megmondja, hogy milyen variációk lehetségesek. Ha valamelyik utasításról vagy programról több információt szeretnék, ha csak beírom, hogy *man* és az utasítás neve, ha pedig nem találok valamit a winchesteremen, akkor a *locate* néhány másodperc alatt kiböki, hogy melyik könyvtárban van az.

Lehet ide-oda mászkálni, nincs külön C:\, D:\ vagy E:\, hanem pl. a CD-ROM a /cdrom könyvtárban van „felmountolva” (ezt a szót is meg kellett tanulnom). Aztán a legszebb az egészben, hogy 256 betű hosszúságig mindenféle file-neveket megadhatok, ja és most jut eszembe, hogy nem kell rettegmem a 640 Kbyte-os megmóriakorláttól, mert itt ilyen nincs, és az a bizonyos „swap partíció” gondoskodik arról, hogy ha esetleg kifogyne a memóriából, akkor is legyen a programoknak hova terjeszkedniük.

Az első napok vegyes érzelmekkel teltek. Beléptem egy új világba, illetve – akkorra már megtanultam – egy új operációs rendszerbe. Először semmit sem találtam, amit begépeltem, az nem működött. Kerestem a futtatható programokat, mert itt nincsenek kiterjesztések, s nincs .com és .exe, ráadásul minden programot összeborítottak különféle bin és sbin nevezetű könyvtárakba, s nem alkalmazásonként lettek szétválogatva a dolgok.

Olyan érzésem volt, hogy újra kell tanulnom a számítástechnikát. Már eleve miféle dolog az, hogy egy PC-n nem csak a DOS és Windows futhat, hanem egy másik operációs rendszer is (megjegyzés: sokáig az OS/2-ről is azt hittem, hogy az valami DOS-os program). Átláthatatlan káosznak véltem az egészet, amíg rá nem jöttem, hogy itt mindennek megvan a helye: a futtatható file-ok a bin könyvtárakban vannak, melyek mindegyike benne van az elérés útvonalban, így ha egy programot le akarok futtatni, csak be kell pötyögnöm a nevét,

sőt elég csak félig bepötyögni, mert ha [TAB]-ot nyomok, akkor a többi jön „magától”. Aztán a fontosabb parancsok többsége csak 2-3 betűből áll, amit ráadásul megjegyezni se nehéz.

Szóval teltek múltak a napok, s egyre többször találtam magamat a Linuxban, s egyre kevesebbszer kellett felhívni ismerősömet, ha valami nem ment. Egyszer aztán elérkezett a nap, amikor felraktam a lilo nevezetű Linux-indító programot a winchesteremre úgy, hogy elsőre az jöjjön be. Kezdtém hanyagolni a Windowst, s az Internetet is már csak Linux alól használtam. Szépen belemelegedtem a Linuxba, már egész jól kiismertem magam a könyvtárak felépítésében, egész jól tudtam már szerkesztgetni a beállításokért felelős konfigurációs file-okat, mígnem egy nap betelt a winchesterem. Annyi mindent töltöttem le a Linuxhoz a Hálózatról, hogy nem volt már hely. Nem volt hát mit tenni, egyszerűen töröltem egy-két DOS-os programot, majd pedig újraparticionáltam a winchestert, s újabb részt formáztam le a Linux file-rendszerévé.

Ma már az 1,5 Gbyte-nyi lemezhelyből mindössze 130 mega a DOS, és nincs már Windows se, letöröltem, hiszen nem használtam már egy ideje, mert végül minden feladatot sikerült megoldanom Linux alól. Ehhez persze időre volt szükség, de a Linux használata során újra felébredtek bennem a programozói ösztönök. A Linux annyira nyílt rendszer, hogy gyakorlatilag minden a gazdája irányítása alatt áll. Tekintettel arra, hogy a linuxos programok többsége az ún. GNU licenz alatt kerül terjesztésre, még a forráskódok is szabadon hozzáférhetőek és módosíthatók. Egyszóval kinyílt előttem a számítástechnika világa, többé nem voltam beszűkítve egy kereskedelmi operációs rendszer lehetőségei közé.

A Linux nemcsak új lehetőségeket, új ismereteket és új barátokat adott, hanem megváltoztatta a gondolkodásmódot is. Rájöttem, hogy a számítógépeket használók többsége nem is tudja, hogy gépe mennyi mindenre lenne képes. Ennek okát abban látom, hogy az egész otthoni és irodai számítógépfelhasználást lényegében egyetlen nagy szoftvergyártó és marketing cég próbálja meg befolyása alá vonni, s egyre inkább törekszik a monopolhelyzet elérésére.

A Linux számomra egy eszme (másoknak vallás is). Elképedve láttam, hogy még van a Földön olyan dolog, ahol nem a pénz dominiál! A Linuxot, illetve a hozzá kapcsolódó szoftvereket olyan programozók fejlesztik, akik ezért szinte soha nem kapnak semmilyen anyagi ellenszolgáltatást, ráadásul

még ők azok, akik saját szabadidejüket és sokszor saját pénzüket áldozzák a Linuxra. Ez egy olyan mozgalom, ahol egyszerűen az összefogás és egymás segítése a jellemző. Ha hiszik, ha nem, vannak linuxosok, akik képesek arra időt és pénzt szentelni, hogy mások gépére ingyenesen felrakják a Linuxot, és ha mások elakadnak, akkor ők segítséget nyújtsanak. Ebben a közösségben egymásért dolgoznak a résztvevők, mert ha valaki megír egy programot, akkor az a másikat elősegítheti újabb programok létrehozására, s ez öngerjesztő folyamat.

Végezetül szeretném megjegyezni, hogy az íráson elején leírt „telepitési kínok” ma már nem így jelentkeznek, mert az újabb linuxos terjesztések (más néven disztribúciók) sokkal barátságosabban és sokkal egyszerűbben tudnak felrakni egy programot, vagy letörölni azt, hogy utána ne maradjon szemét. Gondok, fejlesztésre váró feladatok még most is vannak, sőt mindig is lesznek. De ne legyünk igazságtalanok, amikor valami nem működik, ne szidjuk a fejlesztőket, hanem segítsünk nekik, mert azaz nekünk lesz jobb.

Linux alá is léteznek már kereskedelmi szoftverek az igényesebb felhasználók számára, de gyakori, hogy az ingyenes, Linux-felhasználók által fejlesztett programcsomagok hatékonyabbak és megbízhatóbbak, mint pénzes társaik. A linuxosok legalább ösztínté vállannak programjaikról, a forráskódokba pedig mindenki betekinthat, s javaslatokat tehet a hibák kijavítására.

Én és a Linux tehát jól megvagyunk egymással. Több mint egyéves barátságunk most már gyümölcsseit hozza, mert a Linux munkalehetőséget is terem a rendszergazdáknak, hálózattervezőknek, -építőknél és természetesen a programozóknak. Mielőtt bárki azt hinné, hogy egy évvel ezelőtt én bármit tudtam volna a fenti szakmákról, az téved, egyéves linuxos pályafutásom során tanultam meg minden ilyen ismeretet. A Linuxot nem lehet megugni, olyan mint egy jó könyv, amit nem lehet letenni.

## A barátságos Linux

Mint már írtam, több mint egy éve használó Linuxot és szinte már csak azt. Az elmúlt egy évben megtapasztalhattam a Linux minden kínját és örömet. Kínjaiból tanultam, mert minden gond megoldása külön élmény és tanulság, és örömet nyújtott, mert számítógépem addig rejtett világokat tárt elém.

### HACKEREK VAGY CRACKEREK?

Gyakran hallani, hogy mindenféle gonosz emberek – akiket a tudósítók hackereknek nevez – betörték egy-egy fontos számítógépre, és ott károkat okoztak. Néha azt is lehet hallani, hogy RMS (Richard M. Stallman) vagy Linus Torvalds (a Linux atyja) az utolsó igazi hacker.

Akkor tehát Linus Torvalds azzal szokott szórakozni, hogy mindenféle számítógépekben károkat okoz? Nyilván nem. A félreértés abból adódik, hogy a két hasonló hangzású szót gyakran összekeverik. Pedig a számítógépmániásoknak ezt a két csoportját élesen el kell különíteni.

Mit is jelent a hacker szó? A hacker olyan ember, aki a számítógépek megszállottja. Általában több programnyelven folyékonyan „beszél”. Nem feltétlenül programozással tölti minden idejét, hiszen az általa kedvelt rendszereken többnapos programozást meg lehet úszni az eszközök megfelelő összekötésével.

A hackerre az mindenképpen igaz, hogy tevékenységének eredménye a „hack”. A hack lehet egy csillogó-villogó teljes program, vagy akár egy egyszerű módosítás egy programban, amittől az teljesen nem triviális dolgot fog csinálni. Ugyanigy hack lehet a rendszer elemeinek egy nem egyszerű csoportosítása.

Tehát a hacker az, aki ilyeneket (vagy éppen a GNU C compilert, vagy a Linux kernelt) csinálja. Egy hacker méltóságon alulnak tartja, hogy betörjön valahova, noha a technikákat ismeri. Egyszer-kétszer esetleg megteszi, de akkor is csak a jó barát megráfálása a célja, és sosem okoz kárt.

Aki a károkozás kedvéért betör, az a cracker. A cracker egy teljesen más állatfajta, az egyetlen közös a két típusban a számítógépmánia. Sajnos egy laikus magyarázat nélkül mindkét szó meghallásakor a betörőt látja lelki szemei előtt, amint a kibetér résein keresztül beoson szigorúan védett katonai objektumokba. Talán ezért van az, hogy a „hacker” kifejezés helyett egyre jobban terjed a „guru” megjelölés, noha annak egy kicsit más a jelentése (a guru inkább az operációs rendszert buszgatja, míg a hacker az eszközökkel foglalkozik: a guru inkább konfigurál, a hacker inkább programoz).

*Magossányi Árpád*

A Linux azoknak jó, akikben van kreativitás és szeretik a kihívásokat, mert itt aztán lehet alkotni és elrontani is. Sajnos – az öreg linuxosok akárhogy is tiltakozzanak –, a Linux jelenleg még nem való az egyszerű felhasználó kezébe, mert pontosan sokoldalúságából adódó bonyolultsága miatt csak eltévedne a konfigurációs file-ok és unixos utasítások erdejében.

Amit én szeretnék látni egy linuxos rendszerben a jövőben, az az interaktivitás, a felhasználóbarát, segítő viselkedés, mert ahhoz, hogy minden linuxos ezen nagyszerű közös műve a szélesebb rétegekhez is eljusson, ezeket a feltételeket is teljesítenie kell. Ez azonban már nem rendszerszintű feladat. A rendszer – a Linux-mag – fejlesztői már mindent megtettek azért, hogy most jöhessen egy újabb generációs fejlesztőbrigád, amely a falakra felhúzza a vakolatot is.

Miközben tehát a rendszer tudása nő, egyre jobban fejlődik és elkápráztató dolgokra képes, itt az ideje, hogy a külsőn is alakítsunk egy kicsit. Nagyszerű kezdeményezések a csomagkezelők (hozzá kell tenni, hogy a RedHat vagy Debian csomagkezelőihöz kevés más operációs rendszerben van hasonlóló), ezek már hatalmas terhet vesznek le a kevésbé kísérletező vagy hozzáértő felhasználók válláról, de a profik számára is jobban áttekinthetővé teszik az egész operációs rendszer állapotát.

A programok fordítását/felrakását végező scriptek is egyre intelligensebbek (s nem nélkülözik időnként a humort sem). Ilyen scriptek vizsgálódhatnak jelenlegi beállításaink között, és eredményének megfelelően ajánlják fel a lehetőségeket. Ezen scriptek gyakran már nemcsak szövegesek, hanem dialógusablakkal is megjelennek konzolon vagy az X11 grafikus felületén, mely utóbbi egyenes út a GUI, a teljes grafikus felhasználói kapocs megteremtése felé.

Elsődleges tehát az interaktivitás, ugyanis hiába vannak megjegyzésekkel, magyarázatokkal jól teletűzdelt konfigurációs file-ok, ha a felhasználó gyakran azt sem tudja, hogy merre keresse az általa beállítani kívánt dolgokhoz a megfelelő file-okat. Ezért

aztán menübbe rendezett, interaktív és intelligens programok, scriptek segíthetnek, aztán a felhasználó később eldönti, hogy akar-e mélyebbre menni, vagy megelégszik ezen programok és scriptek által felajánlott lehetőségekkel.

Véleményem szerint ugyanis a Linux sokkal több mindenre képes, mint néhány nagyobb szoftvergyártó operációs rendszere, csak egyszerűen nem elég közvetlen. Természetesen vannak már kezdeményezések mind a kereskedelmi, mind pedig a GNU oldalon. Már Linuxhoz is lehet vásárolni pénzes munkaasztalt (pl.: Caldera Network Desktop), és alakulnak az általam is ismertetett interaktív programok. Ilyen kis grafikus segédletek már találhatóak pl. a RedHat Linux terjesztésben. Persze még ezek is mind fejlesztésre szorulnak.

A barátságos Linux irányába tehát már elindultunk. Egyre több szoftvert írnak X11 alá, vagy meglévő szöveges (parancssorból indítható és paraméterezhető) programhoz alkotják meg a grafikus felületet, gyakran a tcl/tk nyelv segítségével. Ezáltal az egykezes-egeres klikkeléshez szokott felhasználók is jobban érezhetik magukat annál, mint ha egész könyvtárstruktúrákat kellene végigtúrniuk és mindenfélét gépelgetniük.

A Linux mint operációs rendszer – akármennyire hihetetlen – elég jól dokumentált, de (mint az Internetre általában jellemző – a Linux ugyanis az Internet gyermeke finn apától) többnyire csak angol nyelvű dokumentáció érhető el hozzá. Szerencsére a sok linuxos önkéntes között vannak olyanok, akik a dokumentációk, illetve GNU programok helyi változatait készítik. Ez a folyamat már elindult Magyarországon is. Egyre több leírás jelenik meg magyarul, illetve egyre többet fordítanak le angolból. A GNU fordítóbrigádnak már létezik hivatalos magyar tagozata is, így hamarosan kedvenc GNU programjaink magyar változatát is láthatjuk.

A Linux-felhasználók közössége örömmel vár mindenkit, aki részt szeretne venni akár interaktív scriptek, akár magyar dokumentációk, programok írásában. ■



Magossányi Árpád  
(mag@bunuel  
.tli.mata.v.hu)

# A szabadszoftver- mozgalom és a GNU

**Sokan értetlenül állnak azon tény előtt, hogy a bárki által megszerezhető ingyenes programok gyakran jobb minőségűek és kényelmesebben használhatók, mint súlyos pénzekért árult társaik.**



Vannak, akik annyira értetlenül állnak ezen tény előtt, hogy el sem hiszik, pedig ezt többszörösen bizonyították már. Cikemben ennek a miérettjét és hogyanját szeretném megvilágítani.

Először is tisztában kell lenni azzal, hogy az ingyenes programoknak több csoportja is létezik. Vannak olyan programok, amelyek bizonyos feltételek mellett ingyenesen használhatók, de ha valaki komolyan használni akarja őket, akkor fizetnie kell érte. Ezek az úgynevezett shareware programok. Ilyen például az Arj és a StarOffice is. A shareware programok nagy részénél a forráskód nem áll a felhasználók rendelkezésére, de találhatunk olyat is, amelynél igen. Ilyen például a méltán népszerű xv nevű képmasszírozó program.

Vannak olyan programok is, amik ingyenesen használhatók bárki számára, de forráskódjuk nem elérhető, és gyakran a szerző is ismeretlen. Szerencsére az ilyen programok kihalófélben vannak; nagyrészt ezeknek volt köszönhető az a nézet, hogy „az *ingylenprogram* = rossz program”.

A legfontosabb csoportot a szabad szoftverek képezik: a szabad szoftver olyan program, amely nemcsak ingyenesen terjeszthető, hanem a forráskódja is bárki számára szabadon hozzáférhető. Ezek a programok általában a BSD vagy a GPL fedőnevű szerzői jogi védelem alá esnek. A BSD programokat bárki használhatja, másolhatja, módosíthatja, csupán az a kikötés, hogy a tulajdonos neve és a „copyright notice” maradjon rajta a programon. A GPL programok

annyival „tudnak többet”, hogy a szerzői jogi kérdéseket aprólékos módon szabályozták, vagyis ha egy program már egyszer GPL alá esik, akkor annak minden „leszármazottja” is GPL alá esik. Tehát a program használói is biztosak lehetnek abban, hogy a jogaik mindig megilletik őket. Ugyanis volt már rá példa, hogy egy olyan programot, ami addig szabadon hozzáférhető volt, hirtelen kivontak a forgalomból. Sajnos az eredeti Unixszal is ez történt, sokak bánatára. Ahogy a régi (még a Linux előtti időkből származó) nóta mondja:

*Oh Lord, won't you buy me a 4BSD?*

*My friends all got sources, so why can't I see?*

*Come all you moby hackers, come sing it out with me:*

*To hell with the lawyers from AT&T!*

## Miért jók?

Miért mondtuk, hogy a szabad szoftverek sokszor jobbak a pénzért kapható társaiknál? Nagyon fontos szempont az, hogy a szabad szoftver alkotója legtöbbször nem pénzért, hanem a maga használatára vagy szórakoztatására készíti el programját. Ez egyfelől azzal jár, hogy a maga számára legkényelmesebb felületet választja – mivel ő is használja a programot –, másfelől az apróbb kényelmetlenségeket is kiküszöböli. Ezenkívül a program „tesztelése” annak egész életciklusán át tart; amint valami apró hiba felmerül, azt azonnal kijavítja. Fontos az is, hogy a szabad szoftverek íróit a legritkábban szorítja határidő, így jobban átgondolt, biztosabban működő programot tudnak írni.

Laikusok számára talán nem szükségtelen megjegyezni, hogy két programnál, amelyek alapesetben pontosan ugyanazt a dolgot csinálják, kivéve, hogy a változat nem figyeli a hibákat és váratlan körülményeket, a *b* változat pedig igen, a *b* változat nemritkán másfélszer-kétszer annyi dolgot végez el. (Ebből nem szabad levonni azt a következtetést, hogy „nagy program = jó program”, ugyanis is is megfigyelés, hogy egy program, ha alapvetően új tulajdonságot nem tesznek bele, egy kódtisztítás so-

rán igen sokat karcsúsodik.) A szabad szoftver írója ráadásul sokkal jobban fog ügyelni forráskódja kinézetére, logikus szerkezetére is, hiszen biztos lehet benne, hogy programját sokan fogják olvasni. A forrás logikus felépítése általában magával vonja a program logikus felépítését is...

## Ha szabad a forrás

Amiről nem volt szó, az a forráskód terjesztésével járó előnyök. Az Olvasóban felmerülhet a kérdés, hogy miért hangsúlyozzuk annyira a forráskódhoz való szabad hozzáférést? Egyszerűen azért, mert ez a programok jó minőségének a biztosítéka. Ha egy programot sokan használnak, akkor hamar kiderülnek a hibái (mint köztudomású, minden programban végtelen számú hiba van, ez teljes indukcióval könnyen bizonyítható...), és ha a forráskód szabadon hozzáférhető, akkor nagyon valószínű, hogy lesz közülük valaki, aki képes és hajlandó az adott hibát kijavítani.

A tapasztalat megmutatta azt, hogy ha egy szabad szoftver használóinak száma eléri egy bizonyos „kritikus tömeget”, a szoftver elkezd önálló életet élni, és annak az embernek, aki karbantartja, csupán annyi dolga marad, hogy a mások által beküldött módosításokat ellenőrizze, és beletegye a szoftver „hivatalos” disztribúciójába. Akik számára új ez a szabadszoftveresdi, nyilván nem értik, hogy honnan kerül elő az az ember, aki karbantartja, és legfőképpen miért csinálja ezt. A programokat gyakran az írójuk szokta karbantartani, hiszen nemegyszer az adott program az ő mindennapos munkaeszköze is egyben. Sokszor egy csoport alakul arra a célra, hogy egy adott programot karbantartson. Egy ilyen csoport tagjai a legkülönfélébb motivációk alapján végzik ezt a munkát; van aki azért, hogy így mutassa meg a világnak, hogy milyen okos is ő, van akinek szüksége van az adott programra, vagy annak egy adott tulajdonságára, van aki tudományos kutatása közben javítja a programot, azt használva fel kísérleti terepként, és egyre nő azoknak a száma, akik garanciát és támogatást vállalnak egyes cégek felé.

## A kihívás

Itt érkezünk el az informatika mint iparág által nyújtott legnagyobb kihíváshoz: programok egész csoportját ma már nem érdemes pénzért gyártani, hi-

szén szabad megfelelőjük mindent tud, ami az adott programtól elvárható. Ez egyelőre az operációs rendszerek piacán figyelhető meg, igaz ott is csak nyomokban, de a tendencia erősödik. Persze lehet egy ideig erőltetni a dolgokat, hiszen a marketing erős eszköz, amit a szabad szoftverek nem nagyon bírtokolnak, de amikor egy operációs rendszer elkészítésének háromszorosába kerül a marketing, és az elkészült operációs rendszert ezek után is mindenki szidja, a folyamat vége előre látható.

A szabad szoftverek rohamos elterjedése azt hozza magával, hogy az informatikai tevékenység hangsúlya a szoftvergyártásról a rendszerintegráció, a management és support tevékenység felé tolódik el. Bármilyen hihetetlennek tűnik első pillantásra, ez az a terület, amiben a szabad szoftverek verhetetlenek, hiszen egy programról a forráskódja a hozzáférhető számára ékeesebben beszél akármennyi dokumentációnál. Ugyanakkor azt sem szabad elfelejteni, hogy a szabad szoftverek nagy része sokkal többféle környezetben képes futni, mint kereskedelmi társaik. A sok különböző környezethez való alkalmazkodáshoz viszont alapfeltétel a tiszta, áttekinthető kód és a program logikus felépítése; az ehhez szükséges kódírtástól egy szabad szoftvernél sokkal könnyebben elvégzik – hiszen fejlesztőiket nem hajtják a határidők –, mint a kereskedelmi terméket előállító társaikét. Fontos elem az is, hogy a szabad szoftverek fejlesztői általában a hálózaton jobban hozzáférhetőek, nem kötik őket a formások, így válaszüjük sokkal rövidebb. Erre szép példa a nagyon sok operációs rendszert érintő „ping o' death” nevű hiba. *(Ennél a hibánál egy nem szabványos formájú ping csomag hatására számos operációs rendszer képes volt hibás működést produkálni – a szerk.)* Ez a hiba a Linuxnál a felismerés utáni egy napon belül kijavításra került, más – kereskedelmi – operációs rendszereknél erre gyakran fél év nagyságrendű időt kell várni. A felsorolt tulajdonságok összessége teszi a szabad szoftvereket annyira alkalmassá arra, hogy velük hatékony rendszereket lehessen építeni, és ezeknek a támogatását jól el lehessen látni.

## Az emberiség közös kincse

Idáig a tényeket és a folyamatokat mutattam be. Most lássuk a mozzalomban mögötti filozófiát is. Az egész dolog alap gondolata az, hogy a programok forráskódja olyan tudást jelenít meg, amelyet az emberiség közös kincsévé kell tenni. A tudás

ugyanis annyiban nem hasonlít mondjuk a kőolajhoz, hogy míg a kőolaj egyszerre egy ember tulajdona lehet, és ha azt valaki másnak átadja, azal ő szegényebb lesz, addig a tudás átadója a tudás átadása által nemhogy szegényebbé, hanem gazdagabbá válik. A szabad szoftvereknél ez a gazdagodás nyilvánvalóan fellelhető, ha a program forráskódját közrebocsátják, a program szerzője – aki nyilván használja is programját – úgy válik gazdagabbá, hogy a programja mások segítségével még jobban fejlődik. A fenti alapkijelentést egészen odáig lehet élezni, hogy az, aki egy szoftver forráskódját (és ez nem csak itt, hanem bármilyen információval kapcsolatban igaz) nem teszi szabadon elérhetővé, az bünt követ el. A bünt abban az értelemben értem, hogy a társadalom ezáltal olyan mértékű hátrányba kerül, ami nem magyarázható azzal az előnnyel, amihez a „bűnös” ezáltal hozzájut. A hátrány ezáltal éri a társadalmat, hogy az informatika fejlődése megtorpan: mivel egy adott feladatra sok programot kell fejleszteni, hiszen egyik program sem tökéletes (egyrészt azért, mert ezek, mint minden program, természetüknél fogva nem lehetnek tökéletesek, másrészt azért, mert nemcsak a programfejlesztés módja, hanem marketing szempontok sem engedik meg azt, hogy ezek a programok megfelelő módon legyenek megírva és tesztelve). Az alkotó energiáknak nagy részét lekötik ezeknek a duplikátumoknak a fejlesztése: már jól megoldott feladatokat kell még egyszer megoldani, ahelyett hogy az informatikai társadalom az új kihívások megválaszolásával foglalkozna. Ezért van az, hogy a szoftverek fejlődésének üteme lemaradt a hardverekétől.

## A Szabad Szoftver Alapítvány és a GNU-projekt

Ezt a problémát ismerte fel Richard M. Stallman, aki megalapította a Free Software Foundation (Szabad Szoftver Alapítvány) nevű szervezetet. Stallman célja egy teljes, jól használható, bárki részére hozzáférhető operációs rendszer kiépítése volt. Az operációs rendszert GNU-nak nevezte el. A GNU egy rekurzív akronima, azt jelenti, hogy a GNU nem Unix. A rekurzív akronimák használata a hackerek (akiket *nem szabad* a crackerekkel – a betörőkkel – összekeverni) körében általános olyan programok megjelölésére, amelyek egy jól ismert program kiváltására készültek.

Talán segíti a rekurzív akronimák jobb megértését az Emacs esete. Ezt a méltán népszerű szövegszerkesztőt (amit szintén Stallman írt), többen megpróbálták – sikertelenül – túlszárnyalni. Az egyik ilyen próbálkozás neve EINE volt. Az EINE az „Eine Is Not Emacs” (az Eine nem Emacs) rövidítése. Ennek a programnak az ellenpontjaként elkészült egy újabb ZWEI nevű program, amelynek teljes neve a következő: „ZWEI Was Eine Initially” (A ZWEI eredetileg EINE volt). A GNU rendszerre azért is ráillik az, hogy „nem Unix”, mert a rendszer magja nem a hagyományos Unix-rendszerek monolitikus felépítését követi, hanem a mikrokernél-architektúrára épül. Ezt a magot HURD-nek nevezik, és egész sor olyan dolgot tud, amit a unixok általában nem. A HURD egyelőre fejlesztési stádiumban van, kipróbálása csak a nagyon merészeknek ajánlott.

A GNU rendszer nagyon sok eleme viszont teljesen készen van, és más Unix-rendszereken erőteljesen használják is. Ilyen például a GNU C compiler, ami a világ legerterjedtebb, egyik legjobban optimalizáló és legtöbb platformon futó C fordítója. Nagyon sok olyan GNU programot lehetne felsorolni még, amely a kategóriájában a legjobb. A profi Unix-rendszeradminisztrátorok gyakran lecserélik „gyári” programjaikat a GNU megfelelőkre, mert azok kényelmesebbek és jobbak. A Linuxon használt eszközök nagy részét is a GNU rendszerhez írtak teszik ki.

## A GNU General Public Licence

A GNU rendszerhez egy sajátos szerzői jogi iromány, a „GNU General Public Licence” (GNU általános közegedélykelti lehetne fordítani) vagy GNU GPL tartozik. Ez aprólékosan szabályozza a hatálya alá eső programok szerzői jogi helyzetét annak érdekében, hogy a program és annak minden leszármazottja forráskóddal együtt hozzáférhető legyen bárki számára. Tehát az elterjedt hiedellel szemben ezeknek a programoknak legfontosabb tulajdonságuk az, hogy ingyenesek. Az ingyenesség csupán eszköz annak érdekében, hogy biztosítani lehessen a program forrásának hozzáférhetőségét. Ezeknek a programoknak tehát igenis helyük van a szoftverpiacon; a különbség az, hogy nem magukért a programokért fizet az, aki megveszi őket, hanem vagy az adathordozóért, vagy pedig a programhoz járó támogatásért és garanciáért. Hiszen bármilyen jó programok is lennének ezek, nem sok értelmű lenne, ha csupán játkérszerként lennének használhatók. ■

*Horváthné  
Harmati Szilvia –  
(hszilvia@kvark  
.szif.hu)  
Horváth András  
(horvath@rs1  
.szif.hu)*

# A Linux rövid története

**Különszámunk elkövetkező cikkeiben láthatjuk, milyen ma a Linux. Érdeemes azonban egy kis időt fordítani arra, hogyan is lett ilyené. Természetesen lehetetlen minden részletet nyomon követni, hiszen fejlesztők száza dolgoztak a rendszeren. Ezért ebben az írásban elsősorban a Linux rendszermag (kernel) fejlődését követjük nyomon.**

Ezt is csak főbb vonalaiban tudjuk megtenni. A szemléltetés kedvéért lássuk: a 2.0.30-as kernel *CREDITS* file-ja több mint 200 hacker nevét tartalmazza, akiknek a kódja bekerült a kernelbe.

## A Linux születése, csecsemőkora (0.01–0.10)

A Linux története úgy kezdődött, hogy egy finn egyetemista – Linus Torvalds –, megunta az akkoriban oktatási célokra olcsón beszerezhető PC-s Unix, a Minix költősségeit, hibáit, és elhatározta, hogy megpróbál egy olyan rendszert összehozni, ami saját magának egy jobb környezetet biztosít.

Legelőször a 80386 processzor védett módi (protected mode), feladatváltó (task-switching) lehetőségeivel szeretett volna megismerkedni. Ez kb. 1991 nyarának elején lehetett. A pontos dátumra maga a szerző sem emlékszik, de amikor egyszer megkérdezték, mikor van a Linux születésnapja, azt válaszolta: nem tudja megmondani, de egy e-mail tanúsága szerint július 3-án már a POSIX szabvány után érdekelt, így akkor már biztos futott az alaprendszer.

A program fejlesztése Minix alatt történt, eleinte assemblyben. Az első fázisban kialakuló 0.01-es változat

még semmire sem volt használható, csak egy lépcső volt a továbbfejlesztéshez. Linus Torvalds így ír erről:

„Két hónap telt el az alaprendszer felállításáig, de utána rövidesen lett egy lemezmeghajtóm (amely súlyos hibákkal volt tele, de az én gépemen történetesen működött) és egy kis file-rendszerem. Körülbelül ekkor – 1991 augusztusának végén – tettem közé a 0.01-es változatot: nem volt kicsinósítva, nem volt floppy-meghajtója, és nem sok mindent tudott csinálni. Azt hiszem, soha senki nem fordította le ezt a változatot. De akkor már beindultam, és nem akartam addig megállni, amíg túl nem haladom a Minixet.”

Amikor Linus áttért a C nyelvre, a fejlesztés lényegesen gyorsabbá vált, és olyan nagyratörő tervek fogalmazódtak meg, hogy valaha le lehessen fordítani a GNU C fordítóját Linux alatt. (Ma már csak csodálkozni lehet azon, hogy 1991-ben ez volt a nagy álom, és azóta hol tart a rendszer.)

Ez a legelső változat még nem volt használható: csak Minix alatt lehetett lefordítani, és semmi hasznos funkciója nem volt azon kívül, hogy írója megismerkedett a processzossal.

1991. október 5-én hirdette meg Linus az első „hivatalos”, 0.02-es Linuxot az Interneten. Ekkor már néhány alapprogram futott a rendszeren (pl. a GNU gcc nevű C fordítója, valamint a bash burokprogram), így már el lehetett kezdeni használni a rendszert. Ekkor nem is a rendszer használhatóságának növelése volt a fő cél, hanem a rendszermag fejlesztése. Ezért ekkor nem készültek dokumentációk, installációs csomagok, stb. A Linux ekkor még csak a megszállott hackereknek készült.

Linus ekkor elhatározta, hogy az Interneten keresztül bevonja a fejlesztésbe a szabad kapacitással rendelkező programozókat, és a következő hirdetményt tette közzé a comp.os.minix hírcsoportban:

„Sóvárogsz a Minix 1.1 szép napjai után, amikor a férfiak igazi férfiak voltak, és mindenki maga írta a saját eszközmeghajtóját? Egy szép project nélkül vagy, és épp fened a fogad egy operációs rendszerre, amit igényeidnek megfelelően alakíthatsz? Frustrálóan találd, ha minden működik Minix alatt? Akkor ez a levél lehet hogy pont neked szól.

Ahogy egy hónapja említettem, egy szabad Minix-szerűsége dolgozom AT-386 számítógépre. Végül is elérkeztem egy olyan állapothoz, amikor ez egyáltalán használható (bár ez függhet attól, mit akarsz), és a program forráskódját szélesebb körben tervezem szétosztani. Ez még csak a 0.02-es változat, de sikeresen futtattam a bash, gcc, gnu-make, gnu-sed, compress, stb. programokat alatta."

Megjegyzendő, hogy ekkor, és még egy darabig a Linux erősen kapcsolódott a Minixhez: önállóan nem is létezett, csak alatta lehetett lefordítani, futtatni, továbbá az Interneten is a Minix hírcsoportjában folyt a Linux fejlesztésével kapcsolatos információcsere.

A 0.03-as verzió két-három hét alatt megszületett, majd 1991. decemberében Linus kibocsátotta a 0.10-eset is. Ez az ugrás a számozásban azt tükrözte, hogy jelentősen megnőtt a Linux alatt futtatható alkalmazások száma, de a Linux még mindig nem volt önálló, szerzője szerint is „egy hacker által hackereknek írt” rendszerről van szó, így a rendszernek csak fejlesztői vannak, felhasználói nem.

## A Linux gyermekora (0.11–0.99)

1991. december 19-től, a 0.11-es változat kibocsátásától számíthatjuk a Linux gyermekkorát. Ez volt az első önálló rendszer, tehát nem kellett Minix a használatához. Sok olyan tulajdonsággal rendelkezett, amely jelezte, hogy itt valami komoly készül. Ezeket Linus felsorolásában adjuk közre:

„A 0.11-nek a következő újdonságai vannak:

- demand loading;
- kód és adatmegosztás nem kapcsolódó processzek között;
- sokkal jobb floppy-vezérlők (most már többnyire működnek);
- hibajavítások;
- Hercules/MDA/CGA/EGA/VGA támogatás;
- a konzol hangot is ad (Óh! Fantasztikus rendszer-mag!);
- mkfs/fsck/fdisk;
- amerikai/német/francia/finn billentyűzet;
- a com1/2 sebessége beállítható.”

A 0.12-es változat 1992. január 15-én látott napvilágot, néhány bővítéssel: Már volt init/login szolgáltatás (nem rootként kellett először bejelentkezni, és inicializálni a rendszert), közelített a POSIX szabványhoz, virtuális memóriát is használt és kisebb korrekciókat tartalmazott.

Ez már egy elég stabil változat lett, ekkortól kezdődött el a Linux igazi hódítása. A 0.12-es Linuxot ugyan-

is elkezdték használni „egyszerű” felhasználók is, nemcsak megszállokk.

Szintén ehhez a változathoz kapcsolódik a Linux fejlesztésének kiterjesztése: a 0.12-es már lényeges részeket tartalmazott, melyeket nem Linus Torvalds írt. Pl. a job controlt Theodore Ts'o, a virtuális konzolokat Peter MacDonald programozta.

Az így előálló rendszer már több vonatkozásban jobb volt a Minixnél, de még mindig nem volt látható, hogy ebből akkora mozgalom lesz, mint ami mára kialakult.

A sikeren felbuzdulva a verziószám hirtelen ugrott: a 0.95-ös 1992. márciusában, a 0.96 áprilisban lett kibocsátva. Ettől a pillanattól kezdve hihetetlen ütemben gyarapodott a Linux-felhasználók és -programozók száma.

Ekkor a verziószám hirtelen „befékezett”: 1993. decemberében a verziószám 0.99pl14 volt. (A pl14 a *patch level 14* rövidítése, azaz ez a 14. javított változat.) Bár a 0.95-ös verziótól kezdve a szolgáltatások száma, a megbízhatóság, és más egyéb szempontból jelentős javulás következett be, ráadásul hihetetlenül sokan használták ezeket a rendszermagokat, az 1.0 verziószámot mégis csak akkor merték kiadni (1994. elején), amikor a POSIX szabvánnyal való kompatibilitás kielégítővé vált.

A 0.95–0.99 rendszermagra épülő rendszereknek óriási népszerűségük volt. Egyes egyetemeken, pontosabban azok bizonyos intézeteiben gyakorlatilag likvidálták az összes nem linuxos programot, és a PC-ken nem lehetett DOS-t vagy Windowst találni. (Legfeljebb a titkárságokon.) Ez főleg olyan helyeken következett be, ahol a kutatók Unix alatt dolgoztak, mert egy linuxos PC-n otthon is fejleszthették a programjukat, és ezeket egyszerű volt az intézet nagy kapacitású gépeire áttenni.

(Egyikünk személyesen tapasztalta ezt 1993-ban a Würzburgi Egyetem Csillagászati Tanszékén: szó szerint senki sem használt DOS-t, még otthon sem, legfeljebb a kedvenc játékprogramjának futtatására. Az otthoni linuxos gépen kifejlesztett, tesztelt programok gond nélkül áttehetők voltak az intézet CRAY gépeire.

Hazánkban ez az idő tájt (1993) kezdett igazán elterjedni a Linux, mert ekkorra kötötték be a felsőoktatási intézmények nagy részét az Internetre, így sokaknak megnyílt a lehetősége a Linux beszerzésére.

A Linux és a Minix szétválását meggyorsította, hogy a Minix szerzője, Andrew Tanenbaum nem nézte jó szemmel a Linuxot. Alapvetően elhibáztak tartotta a Linux rendszermag típusát; Linus Torvalds ugyanis ún. „monolitikus kernelt” írt, míg Andrew Tannenbaum (elméleti megfontolások alapján) a mikrokernelt jobb választásnak tartotta. Sajnos, a vitába személyes elemek is keveredtek, és a vita az Internet nyilvánossága előtt

zajlott, így kissé ideges hangulatban zajlott le a Minix és a Linux szétválása. Csak izelítőtül:

Andrew Tannenbaum: „Továbbra is fenntartom azt, hogy 1991-ben monolitikus kernelt tervezni alapvető hiba. Örülj, hogy nem vagy a tanítványom. Nem kapnál túl jó minősítést egy ilyen tervezésre :-)”

Linus Torvalds: „Az Ön foglalkozása professzor és kutató: ez egy pokolian jó mentség a Minix némely agysérülésére.”

Andrew Tannenbaum (1992-ben): „A Linux elavult!”

Mivel nem ismerem a Minixet, ezért nem tudok döntőbíró lenni a vitában, de mára a Linux-felhasználók száma, az alkalmazások sokrétűsége és egy csomó más szempont szerint nagyságrendekkel verí a Minixet. Talán elég azt megjegyezni, hogy a Linuxszal kapcsolatos angol nyelvű hírcsoportok száma több mint 10, és magyarul is jó pár (kb. 6) linuxos hírcsoport működik, míg a Minixszel csak egy foglalkozik az egész világon, melynek fogalma össze sem vehető még a magyar Linux-listákéval sem.

Annyit azonban meg kell jegyezni, hogy a Minix azért sem terjedhetett annyira, mint a Linux, mert nem szabad terjesztésű. Ennek ellenére a valószínűsíthető, hogy Andrew Tannenbaumnak nem volt igaza abban, hogy a Linux alapvető szervezése teljesen hibás, mert alapvető hibákkal nem lehetne egy rendszer annyira stabil, és nem is terjedt volna el. Talán személyes ambícióit sértette, hogy egy egyetemista az övénél használhatóbb rendszert hozott össze, mely mellett a Minix elhanyagolható szerepet játszik. Mindenestre sajnálatos, hogy ilyen rivalizálásra is sor kerülhet komoly szakemberek között.

## A Linux fiatalkora (1.0.0–1.2.13)

A POSIX szabványosítás megfelelő szintű elérésével 1994 márciusában megjelent az 1.0.0 sorszámú kernel. Ekkortól kezdve egy speciális sorszámozási eljárást vezettek be a fejlesztők: a verziószámot három, ponttal elválasztott nem negatív egész jelzi. Az első a fő verziószám, ami csak a rendszermag lényegét érintő változásoknál vált eggyel nagyobbra. A második szám elég speciális jelentésű: ha páros, akkor stabil, tesztelt kernellről van szó, amit bárkinek ajánlanak használatra, míg a páratlan szám tesztváltozatot jelöl, amit inkább azoknak javasolnak, akik tesztelni, fejleszteni szeretnék a kernelt, és akiknek nem számít, ha a rendszer néha elszáll. A harmadik szám pedig kisebb módosításokkor ugrik egyet.

Ennek megfelelően egyszerre két szálon fut a legújabb verziószám: pl. e cikk írásakor a legfrissebb két

kernel sorszáma 2.0.30, illetve 2.1.46. A stabil verziókba csak olyan modulok kerülhetnek bele, amelyek a fejlesztői változatokban már üzembiztosnak bizonyultak.

Ez a fura sorszámozás lehetővé teszi, hogy az egyszerű felhasználók csak a valóban használható változatokat kapják meg, de közben az esetleg még hibákat tartalmazó fejlesztői változatok is hozzáférhetőek legyenek. A későbbiekben mi csak a stabil verziószámokkal foglalkozunk, mert ezek a „hivatalos” változatok.

Ezen változatok nem hoztak újabb hatalmas áttörést, mert az már korábban bekövetkezett. A fejlesztés során a rendszermag egyre hatékonyabb lett, beépítették a legújabb hardverek meghajtóit (CD-olvasók, PCI-buszok, újabb SCSI-eszközök, stb.). Talán leginkább az 1.2.x-es kernelekkel bevezetett új végrehajtható programformátum, az ELF megjelenését kell itt megemlíteni.

Ebben az időben a Linux alatti felhasználói programok száma nőtt meg hihetetlenül. Míg korábban főleg már meglévő szoftvereket vittek át a Linux alá, ezután már megjelentek azok a programok, melyeket Linux alatt fejlesztettek, és innen vittek át a többi rendszerre. Ekkora már nagy szoftvercégek is elkészítették programjaik linuxos változatát (pl. Maple V, Motif 2.0). Ezek természetesen nem szabad terjesztésűek, és az hogy megéri Linuxra is adaptálni őket, egyértelműen jelzi, hogy a Linux-felhasználók tábora világméretekben is piacot jelent.

Ekkortájt egyre több Linux disztribúció kezd megjelenni, azaz több cég olyan programcsomagot állít elő, amellyel a Linux telepítése, karbantartása sokkal könnyebb, mintha mindenki egyenként gyűjténé be a rendszer részeit. Ezekről lapunkban több cikk is szól.

## Napjaink Linuxa (2.0.0...)

1996 augusztusában jelent meg a 2.0.0 sorszámú rendszermag. Ennek fő újítása a modulok megjelenése volt: a kernel bizonyos részei modulként is elkészíthetők, és ezek a modulok akár automatikusan, akár kézzel betölthetők a memóriába, ahonnan a rendszer kipakolja őket, ha régóta nem használjuk.

Például a nyomtató, floppy-vezérlő, nem linuxos file-rendszereket kezelő részeket célszerű modulba tenni, mert ekkor ezek csak addig foglalják a memóriát, amíg éppen használjuk őket, és ez többnyire a munkaidő jelentéktelen része.

Ezzel az érdekes helyzet állt elő, hogy a rendszermag memóriagigénye *kisebb* lett, míg hatékonysága és megbízhatósága megnőtt. (Erről a tendenciáról példát

vehetnének a piacot uráló szoftverhatalmak is. Csak-hogy azok nem a hatékonyabb, hanem a nagyobb anyagi hasznot hozó rendszer írásában érdekeltek, és a hardvergyártókkal való összefonódás miatt sokszor a nagyobb memóriai igény kifejezetten előnyös a cég számára. Az is megemlítendő, hogy ekkorra már mintegy 200 hacker kódja került be a stabil kernelbe, és Linus Torvalds idejének jelentős részét ezen munka koordinálása köti le. Az, hogy ilyen szétszórt fejlesztés ellenére a rendszer stabil, működőképés, legalább akkora érdeme Linusnak, mint az, hogy elindította az egész fejlesztést.

## A Linux és a GNU kapcsolata

Kiadványunk egyik előző cikkében kitértünk a GNU és az FSF történetére is. Mindenképp meg kell említeni a Linux és ezen szervezetek kapcsolatát, mert ez döntő volt a Linux fejlődése szempontjából.

Egyrészt, a Linux rendszermag (az első néhány verzió kivételével) a GPL (GNU Public License) hatálya alá esik, másrészt a Linux rendszerek alapprogramjai és a felhasználói programok jelentős hányada a GNU project keretében készült, vagy írója a GPL-t alkalmazza.

Bizonyos szempontból a Linux kiegészíti a GNU projectet. Mint olvashattuk, a GNU egy teljes rendszer szeretne lenni, de ez ideig még nem írtak rá megbízható rendszermagot. (Jelenleg a GNU rendszermag, a Hurd, a 0.2-es verzióval tart, de ez még nem teljesen stabil változat.) Linus Torvalds és társai viszont pont ezt tették meg, igaz, nem a GNU keretein belül.

Nem látható pontosan előre, hogy hogyan alakul a jövő. Az is lehet, hogy a GNU rendszermag nem fog elkészülni, vagy legalábbis elterjedni, mert a Linuxot néhez lesz túlszámolni. De az sem kizárható, hogy Richard Stallman egyszer csak mégis előáll a GNU kernellel, ami lesöpri a Linuxot a színről. (Bár ezt kevésbé tartom valószínűnek.)

## A Linux jövője

Nehéz és veszélyes dolog jóslásokba bocsátkozni, de néhány alapvető dologban biztosak lehetünk:

A Linuxnak mára akkora táborra van, hogy még Linus Torvalds és a többi vezető programozó esetleges kilépése esetén is tovább fog folytatódni a munka. Különösen igaz ez azért, mert mára nemcsak IBM PC-ken, hanem a legerterjedtebb munkaállomásokon is fut Linux, és ezen Linux-változatok némelyikét nem is Linus Torvalds koordinálja.

A mai napig folyamatosan növekszik a linuxos szoft-

verek, a felhasználók száma, és az alkalmazások sokrétűsége. Hogy csak egy példát említsék: egyetemeken és kutatóintézetekben egyre inkább terjed az a szokás, hogy linuxos PC-ket hálózatba kötnek, és az így előálló hardvert valamilyen párhuzamos programfejlesztési környezet alól használják. Az ilyen rendszereknek sokkal kedvezőbb az ár/teljesítmény aránya, mint a készen kapható kisebb szuperszámítógépeké.

Úgy néz ki tehát, hogy lendületben van a rendszer, és több évig tartó biztos dinamikus fejlődés előtt áll.

Véleményem szerint az egyetemi és kutatási helyeken méltó vetélytársa bármelyik ún. „operációs rendszernek”, és itt a jelenleginél sokkal nagyobb elterjedésre számíthatunk a közeljövőben. (Különösen, ha figyelembe vesszük a magyar felsőoktatási intézmények anyagi helyzetét.)

A rendszer stabilítása miatt már most is sok helyen alkalmazzák hálózati szervernek a linuxos gépeket, de még itt is nagy lehetőségek rejlenek a Linux terjedése előtt.

Ugyanez nem mondható el az irodai alkalmazásokról, mert hiányoznak az ehhez szükséges szoftverek, de leginkább a közérthető, több nyelven elérhető dokumentációk. Igaz ugyan, hogy az utóbbi időben ez is változni látszik: pl. a StarOffice és az Applixware rendszerek, a linuxos Word Perfect méltó vetélytársai a mostanáig piacot uráló irodai szoftvereknek.

Az biztos, hogy Linus Torvalds sem lesz a fejlődés gátja. Egy helyen ugyanis ezt írja:

„Azon a napon, amikor a Linuxot valaki más jobban tudja szolgálni (az FSF egy természetes alternatíva), félreállok. Nem gondolom, hogy emiatt valakinek aggodnia kellene, és nem is gondolom, hogy ez a közeljövőben bekövetkezik. Szeretek a Linuxszal foglalkozni, még ha ez egy kis munkával jár is, és még senkitől sem kaptam panaszt. (Csak néhány, majdhogynem szégyenlős emlékeztetőt arról, hogy egy patchet elfelejtettem, vagy nem vettem figyelembe, de eddig semmi negatívát.)

Ne vegyétek úgy a fentieket, hogy azon a napon, amikor valaki ellenkezik, abbahagyom: elég keményfejú vagyok – Lasu, aki ezt a vállam felett olvassa, megjegyezte, hogy a fefejú közelebb van az igazsághoz – egy kis mocskolódáshoz. Ha nem lennék az, abbahagytam volna a Linux fejlesztését, amikor a comp.os.minixen nevettségessé tettem. Csak arra gondoltam, hogy bár a Linux eddig az én gyermekem volt, nem akarok útban lenni, ha az emberek jobbat szeretnének csinálni.”\* ■

\* Hé, lehet, hogy kérhetném a szentíé avatásomat a Pápától! Tudja valaki az e-mail címét?\*

*Botka István*  
(*boti@kac.polioid*  
*.hu*)

# Levelezés Marssal

**Főiskolánkon úgy alakult, hogy a hallgatói levelezés adminisztrálása hallgatói kezekbe került. Ennek a történetét szeretném röviden vázolni.**

Az Informatikai Intézet vezetőivel való többszöri konzultáción kiderült, hogy olyan megoldást kell keresni, ami:

- Viszonylag könnyen adminisztrálható.
- Megfelelően biztonságosan működő.
- Az adminisztráció automatizálható legyen, még nagyszámú felhasználó esetén is. (Évente kb. 500-800 új hallgató jön és ugyanennyi távozik.)
- A megvalósítás anyagi vonzata lehetőleg minimális legyen.

Ezek után nekiálltunk kidolgozni különféle alternatívákat, bár már az elején világos volt néhány dolog: a kiszolgáló a kollégiumi Linux-szerver lesz, az adminisztrálást a kollégiumi géptermet üzemeltető szervezettel látja el.

Gép már van, megbízhatóan működik, bár a hardver kapacitása kissé szűkös. Bővíteni azonban lehet. A levelezés konkrét megvalósítására vonatkozólag több változat is volt:

- TCP/IP protokoll felett telnet kapcsolattal, „normál” Unix.

- Szintén TCP/IP felett Pegasus Mail használatával POP3 és SMTP protokoll felett.

- Ha az iskolai munkahelyeken TCP/IP felélesztése nem lenne lehetséges, akkor IPX felett kell megoldani a levelezést.

Végül is ez az utóbbi maradt. Eddig korlátozott számban volt hallgatói levelezés az intézményi Novell Netware 4.10-es szerveren keresztül. A hallgatók többsége ismeri a Pegasus Mailt. Alapfeladat volt annak megoldása, hogy a pmail számára valahogy elő kell készíteni a leveleket olvasásra, és amit otthagya a spool könyvtárban, azt utána továbbítani kell a Sendmailnek. (A Sendmail filozófiája szerint a felhasználó számára érkező levelek egy közös könyvtárban – általában a /var/spool/mail – mindenkinek a saját usernevén egy file-ban tárolódnak. A Pegasus Mailt általában a Mercury nevű levél-átjáró (gateway) programmal alkalmazzák. Ez utóbbi kiszórja a beérkezett leveleket a felhasználók mail könyvtárába, minden levelet külön külön file-ba.) Két programcsomag felhasználá-

sával láttunk neki a feladat elvégzésének: az egyik a mars-nwe, a másik a marsmail. Mindkettő megtalálható a sunsite mirrorokon, ha valakinek van Internet-elérése. (Az URL-ek pl: ftp://ftp.kfki.hu/pub/Linux/system/filesystems/ncpfs/mars\_nwe-0.98.pl8.tgz illetve ftp://ftp.kfki.hu/pub/Linux/system/filesystems/ncpfs/marsmail.tgz). A mars a legtöbb Linux disztribúció része. A kollégium szerverén RedHat 4.1-es fut. A mars felélesztése és konfigurálása nem jelentett gondot, hiszen eléggé jól kommentezett a konfigurációs file-ja (/etc/nwserv.conf). A marsmail pedig „The marsmail is just a dirty hack to satisfy the dos users.” Azaz csak egy hack ahhoz, hogy DOS-felhasználók tudjanak levelezni. A lehetőség ugyanúgy adott a „hagyományos” levelezésre.

A Pegasus Mailnek van egy olyan lehetősége, hogy off-line levelezik, egy felhasználó által definiált átjárót használva. Ez gyakorlatilag annyit jelent, hogy kell biztosítani egy-egy könyvtárat a bejövő és a kimenő levelek részére. Ekkor az új leveleket a pmail indulása előtt egy külső programnak ki kell szórnia az IN könyvtárba, leellenkint jövelel file-ba. Az átjáró konfigurálásához supervisorri jogokkal kell rendelkezniünk a Netware (mars) szerveren, mert az átjáró adatait a binderybe rögzíti a pmail. Magát a konfigurálást a Pegasus Mail csomagból a PCONFIG.EXE használatával tudjuk elvégezni.

A felhasználó a pmaillel elolvassa a fent vázolt módon előkészített leveleit, megválaszolja a megválaszolandókat, illetve elküldi amit szeretne. A pmail ilyenkor a kimenő leveleket csak betesz a átjáróban megadott OUT könyvtárba és szintén a külső programra bízta annak feldolgozását.

A marsmail csomag része egy Pascal program, amit DOS oldalon indít a PMAIL.BAT. Ez a Pascal program (MARSMAIL.EXE) végzi a unixos levélfile szétbontását a pmail számára emészthető formátumra. Előtte azonban a unixos levélfile keresztülmegy egy unix2dos konverteren, ami elvégzi az LF → CR/LF konverziót, azaz a sorvégek „rendbetételét”.

DOS oldalról hozzá kellett igazítani a bejelentkezést a marsmailhez. Ez annyit jelentett, hogy irtunk egy kis programot, ami bekérte a marsos login nevet és a hozzá tartozó jelszót, felléptette a szerverre, elvégezte a megfelelő meghajtók mapelését, és a szükséges környezeti változókat beállította.

Csak ennyi volt az egész, s a rendszer működik! ■



Vid Gábor  
(vidg@geobio  
.elte.hu)

# Linux a hálózat- managementben

**Az itt bemutatásra kerülő számítógépes labor az Eötvös Loránd Tudományegyetemen működő geológus-biológus számítógépes hallgatói labor.**

A labor két különálló helyiségből áll, ez a tény tette lehetővé, hogy a kiszolgáló gépeket biztonságosan elkülönítve tudjuk tartani. A hallgatói labor dr. Kovács József egyetemi adjunktus vezetése alatt áll, az üzemeltetésben korábban Kondorosi Gábor, jelenleg Tóth Krisztián és Vid Gábor hallgatók vesznek részt.

## A labor korábban

A hallgatói labor létrejötte óta eltelt időben a lehetőségek és igények növekedésével lépésről lépésre alakítottuk át a rendszert, fontos szempont volt, hogy a felhasználók ezt minél kevésbé vegyék észre. Jelen cikkben nem szólnunk azokról a fejlesztésekről, problémákról, amelyek nem kapcsolhatók a Linux felhasználásához.

A hallgatói labort egy OS/2-LAN szerver szolgálja ki, ezen a szerveren helyezkednek el azok a programok, melyeket a hallgatói gépekről el lehet érní (elsősorban windowsos alkalmazások). A gépek teljes Internet-eléréssel is rendelkeznek, azaz valamennyi gépnek önálló IP-címe is van (elsősorban telnét, ftp, WWW alkalmazásokra használják őket).

Az OS/2-LAN Serveren a NETBIOS/NETBEUI protokollt használjuk a helyi forgalomnál, az internetes forgalomnál természetesen TCP/IP-t. A két különböző protokollt egy időben használjuk, ugyanazon a fizikai közegen. Mivel a helyi forgalom elég jelentős, ezért már eleve úgy lett kialakítva a hálózat, hogy egy PC-t a hálózat bridge-elésére\* használtunk, ezt egy PC-Bridge szoftver segítségével oldottuk meg. Erre azért volt szükség, mert a hálózati topológia miatt helyi forgalmunk az egész épület (illetve még egy másik épület) hálózatát terhelte volna.

A problémákat ez a PC-Bridge okozta, ugyanis a megbízhatósága igen alacsony volt, ez azt jelentette, hogy gyakran váratlanul abbahagyta a bridge funkciók végzését (azaz a két oldal nem „látta egymást”). Az általunk akkor használt free szoftver semmilyen hibajelzést, semmilyen látható jelet nem adott ilyen esetekben. Ez az állapot a belső forgalom növekedésével romlott, azaz a meghibásodások gyakorisága jelentősen növekedett. Ekkor merült fel bennünk az a gondolat, hogy próbáljunk meg egy PC-ből Linux operációs rendszer felhasználásával routert építeni, amely a forgalomsztérválasztást el tudja végezni.\*\* Ezen elképzelésünk megvalósításához az ELTE Informatiótechnológiai Igazgatóság szakemberétől, Győri Gábortól jelentős segítséget kaptunk.

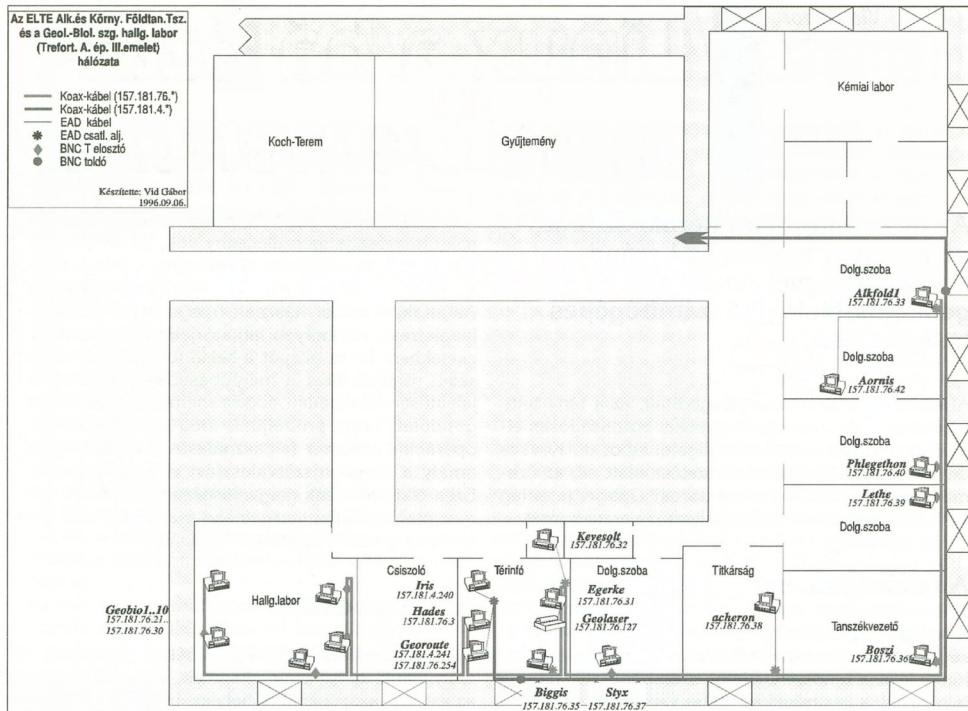
## A Linux bevonul

Az első kísérletek múlt év áprilisában kezdődtek. Miután láttuk, hogy ezek sikeresek, akkor történt meg az átállítás, azaz a bridge helyére betettük az újonnan kialakított routert. Ez a router akkor egy 386 DX40-es alaplappból, 8 Mbyte RAM-ból és 320 Mbyte-os merelemezéből, valamint két darab SMC-ULTRA (8216) ethernet kártyából állt. Ez év januárjától az alaplapot kicseréltük egy P100-asra, valamint a megnövekedett feladatok miatt egy 1 Gbyte-os merevlemezét is beszereltünk. A PC-ből ily módon kialakított router két ethernet szegmens között végezte a route-olást.

Múlt év szeptemberében elkészült a laborral szomszédos, és az ahhoz szervezetenként is igen közel álló tanszék hálózata, melynél már adott volt a lehetőség, hogy a már néhány hónapja üzemelő routert vegzeze a helyi és kimenő forgalom szétválasztását. Ekkor – 1996 szeptemberében –, a tanszéki hálózat kiépítésekor született meg az ábrán látható struktúra.

\* A bridge egy hálózat két szegmense közti forgalom szétválasztását végzi a fizikai címek alapján. A bridge-ben két ethernet kártya van, amelyek minden csomagot fogadnak mindkét oldalon, miközben egy táblázatba feljegyzik a két oldalban használt fizikai ethernet címeket. Ha egy címzett fizikai címéből – ez pedig az ethernet frame első 6 byte-ja – megállapítja, hogy a feladóval azonos oldalon van, akkor eldobja a csomagot, ha az ellentétes oldalon, akkor továbbítja. Ha a címzett fizikai címe még nem szerepel egyik oldali címetáblázatban sem, akkor is továbbítja a csomagot.

\*\* A router ellentétben a bridge-dzsel a forgalom szétválasztását az IP-címek alapján végzi, tehát a belső nem IP-alapú forgalmat semmilyen körülmények között sem engedi át.



Amikor a route-olás már üzemszerűen működött, elvégeztük a sebességek vizsgálatát is. Itt a válaszidők – ping, valamint az ftp idejének – vizsgálatából kiderült, hogy átlagos terhelés mellett nem ad észlelhetően lassabb elérést, mintha közvetlenül érnének el a hálózatot.

## További fejlesztések

Miután a route-olás megvalósítása után láttuk, hogy a gép rendelkezik kihasználatlan erőforrásokkal – bírta a terhelést –, felinstalláltunk egy Web-szervert, és ezen elhelyeztük a hallgatói labor számára lényeges információkat tartalmazó Web-oldalakat. Néhány héttel később levelezési lista készítéséhez szükséges programot is telepítettünk, ezáltal létre tudtuk hozni azt a levelezési listát, amely arra hivatott, hogy az esetleges üzemzavarokról, eseményekről tájékoztassa felhasználóinkat.

Ebben a helyzetben már teljesen egyértelműen adódott az a lehetőség, hogy a labor, illetve az épület számára a nameservice-t is ezen a gépen lássuk el. A nameserver felinstallálása szintén nem igényelt komolyabb munkát, hiszen az ide tartozó program (named) része a Linux disztribúciónak, mindössze a zóna file-okat, valamint a /etc/named.boot file-t kellett megfelelően kitöltenünk. A nameserver beindítása sem okozott jelentős problémát a gép eredeti funkciójában, azaz továbbra is végzi eredeti feladatát, a forgalom szétválasztását és a route-olást.

Amikor az eszközállomány DEC VT-320 terminálokkal bővült, akkor az ezeket kiszolgáló Xyplex terminálszerver tftp bootolását is erre a Linuxra bíztuk. Ekkor került felinstallálásra a tftp szerver és a bootp szerver. Ezzel a bootp szerverrel egy másik probléma is megoldódott, miszerint a hallgatói labor önálló IP-címmel és ethernet kártyával rendelkező lézernyomtatóját is tudjuk bootp-vel konfigurálni – az IP-címét

## ÍGY KÉSZÜL A ROUTER

A Linux kernelszinten támogatja a route-olást, semmilyen egyéb szoftver nem szükséges hozzá, mindössze az erre vonatkozó opciót kell a kernel fordításakor engedélyezni. Ha a kernelfordítás opcióit a *make menuconfig* segítségével állítjuk be, akkor a *Networking options* menüben kell az *IP: forwarding/gatewaving* lehetőséget engedélyezni, valamint be kell állítani, hogy a rendszer két ethernet interface-szel rendelkezik. Ezt a */etc/lilo.conf* file-ban egy *append* paranccsal tehetjük meg.

Az általunk használt ethernet kártyáknál ez a következő: „*append ether=10,0x300,eth0 ether=11,0x280,eth1*”. Az első paraméter az IRQ sorszáma, a második a bázis I/O port címe, végül az utolsó paraméter az, hogy az adott kártya a rendszer melyik ethernet interface legyen.

Ezek után be kell állítani a két interface paramétereit (IP-cím, subnet mask stb.), ezt disztribúciótól függően valamelyik *rc* file-ban kell megtenni. Az általunk használt Slackware disztribúcióban ez a */etc/rc.d/rc.inet1*-file-ban történik. Ez interface paraméterek beállítása után a kernel routing táblát állítjuk be, mégpedig úgy, hogy a default route a külvilág felé mutat, és a belső hálózat felé egy network routingot csinálunk. A disztribúció konfigurációs lehetőségeivel – amit az installálás folyamán kínál fel – a külső (eth0) interface paramétereit állítjuk be, majd utólag kézzel módosítva a megfelelő file-t, a másik, a belső hálózat felé néző interface paramétereit. Mi a következő beállítást használjuk:

```
/sbin/ifconfig eth0 151.118.4.241 broadcast 151.118.4.255 netmask 255.255.255.0
```

```
/sbin/ifconfig eth1 151.118.76.254 broadcast 151.118.76.255 netmask 255.255.255.0
```

A routing táblába default route-nak a külső hálózat routerét állítjuk be, a következő parancsok segítségével:

```
/sbin/route add -net 151.118.4.0 netmask 255.255.255.0
```

```
/sbin/route add default gw 151.118.4.254 metric 1
```

```
/sbin/route add -net 151.118.76.0 netmask 255.255.255.0 eth1
```

Természetesen az itt látható címek helyett mindenkinek a sajátjait kell használnia.

és ehhez szükséges további paramétereket –, ami lényegesen egyszerűbb teszi a nyomtató hálózati paramétereinek állítását.

Az újabb kernelek támogatják a firewall technikák alkalmazását, ennek paraméterezésére az *ipfwadm* programot használjuk. Itt a terminálszerver, a nyomtató, valamint a LAN-szerver IP-forgalmát szabályozzuk. Illetve lehetőség van az átmenő forgalom naplózására is. Jelenleg óránkénti összeget naplózunk mind a bejövő, mind a kimenő forgalomban. Ezeket a funkciókat egy néhány soros script vezérli, a forgalommérést a cron segítségével végezzük.

A gép bővítésével (további merevlemez) lehetővé vált, hogy *nfs*-en keresztül file-szerver funkciókat lásson el. Ezzel valóra vált az a tervünk, hogy az így létrehozott diskról a labor gépei loadlin segítségével *nfs*-en mountolt filesystemmel Linux futtatására alkalmasak legyenek. Az első próbálkozások biztatók.

Nagy előnye ennek a megoldásnak, hogy a hallgatói labor IP-forgalmának, Internet-elérésének menedzselését a LAN-szerverfunkcióktól teljesen szétválasztva tudjuk végezni. Ezzel az üzembiztonságot növeltük, azaz a LAN-szerver meghibásodása esetén a labor és a hozzá kapcsolódó tanszék Internet-elérése

nem szűnik meg, de ez igaz fordítva is, hogy a router leállása esetén a LAN-szerver funkciók sem szűnnek meg, csak a külső Internet-elérés válik lehetetlenné.

## Terveink

Tudjuk, hogy a jelenlegi rendszer még számtalan lehetőséget kínál. Az egyik az, hogy megvalósíthatóvá válik a tanórák alatt a labor gépeinek automatikus leválasztása az Internetről, lehetőségünk lenne arra is, hogy valamennyi hallgatónknak e-mail címet adjunk, amit a laborból POP3 segítségével érhetnének el (ennek megvalósításában csak az az egyetlen akadály, hogy 700 körüli felhasználó levelezésének lebonyolításához kevés a rendelkezésre álló merevlemez-kapacitás). Legközelebbi tervünk az, hogy egy *http proxy* szervert installálunk fel erre a gépre, ettől további átmenőforgalom-csökkenést remélünk. Összefoglalásul elmondhatjuk, hogy a közel egyéves üzem alatt olyan meghibásodásunk, leállásunk, üzemzavarunk nem volt, amely ne valamilyen operátori hibára lenne visszavezethető. A fent leírt rendszer eddig nem okozott csalódást, megbízhatósága megfelelő. ■

Valter Krisztián  
(flex@niki  
.vpszk.bme.hu)

# Dosemu, avagy DOS a Linux alatt

**Az átlagos Linux-felhasználó, miután felinstallálta a Linuxot, elmerül a rengeteg új program és lehetőség tengerében. Sokáig nem is tudja, hogy mi az, ami egy picit mégis hiányzik. Aztán rájön, hogy a jól megszokott DOS-os programok, játékok azok, melyek nincsenek a palettában. De a Linuxon semmi sem lehetetlen.**

A Linux alatt természetesen lehetőség van DOS-os programok futtatására is. E cikk a dosemu lehetőségeiről szól, valamint arról, hogy hogyan vegyük rá Linuxunkat a problémamentes emulációra.

A dosemu egy DOS-emulátor Linux alá. Pontosabban: megfelelő környezetet próbál teremteni a DOS futtatásához. Ezt úgy teszi, hogy készít egy virtuális számítógépet, ott lefuttatja a bootoláshoz szükséges file-okat, és máris DOS-os felületünk van, miközben a Linux többi szolgáltatása továbbra is él. A dosemu foglalkozik teljes egészében ezzel a virtuális géppel, elégti ki az ott indított programok minden igényét, és csak annyit enged látatni mindebből a Linux többi részének, amennyi szükséges, vagy amennyit láttatni akarunk.

## Hogyan kezdünk hozzá?

Ha újonnan telepítettük a Linuxot, és a telepítőcsomag felajánlotta a dosemu telepítését, akkor ugorjuk át a most következő részt, és a dosemu konfigurálásánál folytassuk – ugyanis minden valószínűség szerint a dosemu már ott csücsül a gépünkön, csak még egy kis igazításra szorul.

Az installhoz szükségünk van először is egy Li-

nuxra. Nagyon fontos esetünkben a kernel verziószáma, mivel a 2.0.28-as kerneltől kezdve be van építve a vm86plus szolgáltatás, amelyet a korábbi kerneleknél csak modullal vagy kernel patchéssel érthetünk el.

Kell még egy DOS-os bootlemez, FDISK-kel és SYS-szel, valamint a boot file-okkal.

Nem árt még sok-sok memória, és esetleg egy angol szótár, ha nagy gond van. És persze maga a dosemu forrás, amelyet letölthetünk bármely ismert ftp szerverről, de a CHIP-CD-ken is ott szoktak lenni az aktuális verziók. E cikkben a dosemu 0.65.0 verzióját használjuk, de a korábbiak is közel hasonló módon használhatóak. Ha nem akarunk a régebbi verziók hibái miatt feleslegesen gondot okozni magunknak, akkor 0.64.4 vagy e feletti verziót használjunk.

Első feladatunk a fordítás. A forrás főkönyvtárában elindítjuk a ./configure scriptet, amely átnézi a szükséges paramétereket, és létrehozza a fordításhoz a Makefile-okat. Rengeteg kapcsolója létezik ennek a scriptnek, most csak a 2.0.28 és az e feletti kerneleknél szükséges `-enable-kernelvm86plus` lesz számunkra fontos, mert így közeljünk a dosemuval, hogy kernelünk beépített vm86plust tartalmaz. Ezután a fordítás következik a `make parncssal`, majd a root által kiadott `make install`, mely bemásolja a lefordított file-okat a megfelelő helyekre. Ezzel teendőkön első felével meg is lennénk.

Előfordulhat, hogy 2.0.28-nal régebbi kernelünk van, vagy régebbi dosemuunk. Ebben az esetben szükségünk lesz az emodule nevű modulra, amelyet a forrás főkönyvtárában található `load_module.sh` scripttel tölthetünk be. Ehhez korábban a kernel fordításánál engedélyeznünk kellett a modulok támogatását.

Mostanában merült fel problémaként, hogy a RedHat 4.0-ről 4.1-re való áttérés során hibaüzenettel leáll a dosemu. Ilyenkor három dolgot tehetünk: vagy újrafordítjuk a kernelt, vagy ugyanazt tesszük a dosemuval, vagy a nemrég kijött dosemu patchet használjuk.

A következő lépés a dosemu konfigurálása. Ehhez a forrás /etc/config.dist file-ját kell átmásolnunk a /etc könyvtárba, dosemu.conf néven. A konfiguráláshoz segítséget nyújt ezen kiadványunk CD-jén található *Cikkekhez/dosemu/dosemu.minta* file, melyet az amúgy is bőséges angol kommentezés mellett magyar nyelvű megjegyzésekkel is elláttunk.

A konfigurálás során elérkezünk a HARD DISK menüponthoz, ahol választás elé kerülünk: hogyan is használjuk a dosemut?

Egyrészt létrehozhatunk egy új meghajtót, ún. hdimage-et, amelyen a futtatni kívánt DOS-os programokat tároljuk. Mellé be lehet mountolni akár a Linux, akár az esetleg már meglévő DOS-os partíciót, így ezek is elérhetőek lesznek.

A másik lehetőség, hogy a dosemu a DOS-t a már meglévő DOS-os partícióról fogja futtatni, így a dosemu indítása után a jól ismert config.sys-ünk és autoexec.bat-unk köszön vissza ránk. Addig viszont még akad pár dolgunk. Nézzük meg, milyen lépések szükségesek az egyes változatokhoz.

## Bootolás hdimage-ből

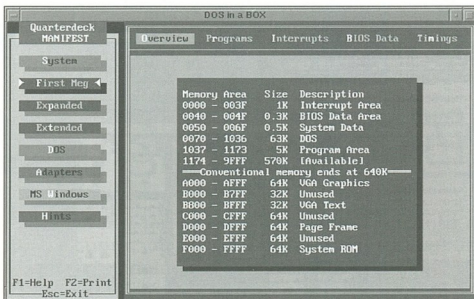
Sok olyan ok van, ami miatt nem a meglévő DOS-os partícióját használja az ember, hanem létrehoz egy image file-t, és ebben tárolja az általa használt DOS-os programokat. Például, ha azt akarja, hogy bárki használhassa a dosemut, biztos nem örülne neki, ha mindegyik felhasználónak elérési joga lenne a teljes DOS-partícióhoz. Mert bizony, ha nem ezzel a módszerrel futtatjuk a dosemut, akkor mindenki a DOS-os partíción keresztül fog bootolni, ezzel engedélyezve azt, hogy egy távoli gépről bármit elkövethessenek ezzel a partícióval. Ennek tükrében talán már látszik, hogy van előnye a hdimage-nek.

Ha e megoldás mellett döntöttünk, akkor a HARD DISK menüpontban a:

```
disk { image "/var/lib/dosemu/hdimage" }
# use diskimage file
```

rész elől vegyük el a „#”-ot, mint itt látható. Ez többi lehetőség előtt persze legyen ott a #. Ezzel közöltük a dosemuval, hogy a hdimage-ről bootoljon.

A hdimage létrehozásához másoljuk át a forrás /etc könyvtárában levő hdimage.dist file-t a /var/lib/dosemu könyvtárba, hdimage néven. Tegyük be a DOS-os bootfloppyt a meghajtóba, majd adjuk ki a „dos -A” parancsot.



Né ijedjünk meg, mert a gépünk úgy tesz majd, mintha bootolna (video BIOS, BIOS bejelentkezés, RAM-ellenőrzés stb.), de csak jelzi, hogy elindult a boot-folyamat a virtuális gépen, másrészt ténylegesen futtatja a számára szükséges programokat.

Rövidesen megjelenik a DOS-prompt. Ezentúl C: néven fut a hdimage-ünk, melyet a DOS dir parancsával meg is nézhetünk. Hogy bootolni lehessen erről a hdimage-ről, a floppyt adjuk ki a az „fdisk /mbr”, majd a „sys a: c:” parancsot. A művelet befejezése után lépünk ki a dosemutból. Ezt vagy a C:-on található exitemu paranccsal, vagy a [Ctrl+Alt+PgDn] gombok megnyomásával tehetjük meg.

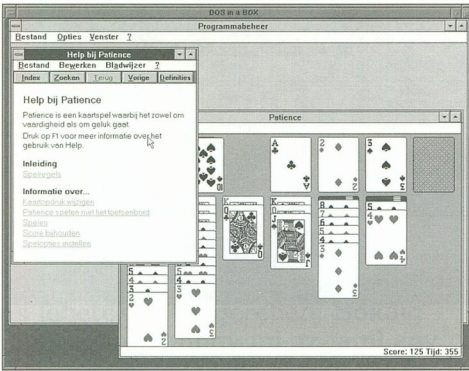
Próbáljuk ki, hogy működik-e az új hdimage-ünk. Írjuk be újra a „dos” parancsot, immár paraméterek nélkül. Ha minden rendben van, akkor megjelenik a DOS-prompt.

## DOS a DOS-partícióról

A dosemu használatának kellemeesebb módja a meglévő DOS-partícióról való bootolás. Itt is természetesen a dosemu saját bootolásáról van szó, ténylegesen nem indul újra a gép. A dosemu.conf-ban a HARD DISK alatt az ehhez szükséges menüpont: disk { partition "/dev/hda1" }

Ezzel tudtára adtuk a dosemunak, hogy a /dev/hda1-ről (vagyis az elsődleges merevlemez első partíciójáról) akarunk bootolni, melyen a siker érdekében nem árt, ha tényleg DOS-partíció van, a szükséges boot file-okkal.

A dos parancs kiadása után megjelenik az ismerős bootfolyamat, majd a CONFIG.SYS és az AUTOEXEC.BAT lefut, és megkapjuk a promptot.



Itt kell megjegyezni, hogy e megoldás választása esetén a DOS-os partíció nem lehet felmountolva, vagy ha mégis, akkor a dosemu futtatása előtt unmountolnunk kell azt.

Mivel a dosemu használatával a Linux sok beépített dolgot ad át a DOS-nak, ezért többek között nem lesz szükségünk HIMEM.SYS-re, EMM386.EXE-re, valamint IPX hálózatról a hálózatkártya driverére és egyéb programokra.

Hogy ne kelljen végignézni a figyelmeztető üzeneteket dosemu alatt, létrehozhatunk config.emu és autoexec.emu file-okat, amelyek ugyanúgy viselkednek, mint hagyományos társaik, de ezekben elég csak azokat a programokat futtatnunk, amelyek szükségesek a dosemuhoz. Ha ezt a megoldást választjuk, akkor a dosemu.conf-ban engedélyeznünk kell az „EmuSys EMU” és „EmuBat EMU” menüpontokat a DOSEMU BOOT-nál.

## Milyen programok futnak a dosemu alatt?

A dosemu fejlesztése nem kezdődött túl régen, és mint a verziószám mutatja, csak tesztverzió létezik belőle. Alakítása és tökéletesítése folyamatosan halad, 3-4 hetente jönnek ki az újabb verziók, de azért ne várjunk tőle teljes kompatibilitást!

Sokféle szempont van, ami meghatározza egy emulátor minőségét. A két legfontosabb, a programok futási sebessége és a futtatható programok száma.

Az előbbiről vegyes vélemények alakultak ki: néhány program, ami erősen használja a processzort, az talán jobban is fut dosemu alatt, mint DOS-ban. A többi program sebességét memória- és processzorigénye alapján osztályozhatjuk. Emiatt memóriából nem árt, ha bőséggel rendelkezünk. A dosemu.conf-ban beállíthatjuk a rendelkezésre álló DPMI, XMS és EMS memória mennyiségét, valamint a maximálisan láttatható konvencionális memóriát.

Viszont ne feledkezzünk el arról, hogy a DOS-os program mellett még maga a Linux is működik, így ha van 16 Mbyte memóriánk, és ebből 13-14-et felhasznál a DOS, akkor bizony már swappal a winchesterünk, és ez lelassítja az egész folyamatot. Ökölszabály, hogy ha sok memóriát használ egy DOS-os program, csak akkor indítsuk el, ha emellett még marad elég memóriánk a Linux működésére, a háttérben futó programok igényeinek megfelelően. Ha nem így teszünk, attól még valószínűleg elindul a program, de kinszenvedés lesz az állandó winchesterberregés és a lassúság.

Persze sebességről csak akkor beszélhetünk, ha elindul egyáltalán a programunk dosemu alatt. A fejlesztés körkezdik a végső stádiumához, bár még vannak sarkalatos kérdések, amelyek megoldásáig nem lehet tesztverzióról átváltani a „végleges” verzió(k)ra. Örvedetes tény, hogy már a mostani állapotban is csak kevés program van, ami nem hajlandó futni. A forrás/doc könyvtárában kapunk egy Emusuccess.txt file-t, amiben a „valakinek már futott” programok listája van. Ez azt jelenti, hogy minden valószínűség szerint nálunk is fog, ha mégse, akkor írhatunk az ott szereplő e-mail címre, amelynek tulajdonosa azt állította magáról, hogy nála bizony futott. Persze vannak olyan programok is, amelyek nem futottak dosemu alatt, és szinte biztosan az emulátor miatt.

Windowsból a 3.0-s verziót standard módban tudja futtatni, ha 3.1-eset akarunk, akkor az OS/2-höz adott Windowszal próbálkozzunk, ahogyan a Readme.Windows file-ban le van írva. Emellett lehetőség van 3.1-es Windows futtatására a Wine vagy a Wabi programok segítségével is, de ezek nem a dosemu részei.

Remélhetőleg ez a cikk segítséget nyújt a dosemu konfigurálásához és a DOS Linux alatti zökkenőmentes használatához. Nehézséget a fordítás és a konfigurálás szokott okozni, ha a dosemu már elindul, akkor a felhasználó rájön egy idő után, mi is kell kedvelt programjainak. Sok sikert, kellems upgrade-et... ■

Lajber Zoltán  
(lajbi@jht.gau.hu)

# SaMBA – kapcsolat a rendszerek között

**Az elkövetkező oldalakon bemutatok egy megoldást arra, hogy hogyan lehet egy linuxos gépet egy munkacsoport számára file-, nyomtató- és faxszerverré tenni.**

Cikkemben ismertetek egy tipikus szerver oldali beállítást, néhány megfontolandó gondolatot a felhasználók jogairól, és persze néhány kliens programról is szó lesz. Ennyi információ csak az elinduláshoz elég, de a SaMBA nagyon jó dokumentációval rendelkezik, persze angolul.

A cikk eredetileg a Magyar Linux Felhasználók második „buliján” elhangzott előadásom módosított anyaga, melynek létrejöttéhez Iványi Péter (peteri@carme.sect.mce.hw.ac.uk) és Szekeres István (szekeres@fornax.hu) nyújtott segítséget.

## Mi is az a SaMBA?

Szabadon fordítva a SaMBA FAQ-ból: A SaMBA olyan programcsomag, amely elérhetővé teszi a kliensek számára a kiszolgáló file- és nyomtatóterületeit az SMB (Session Message Block) protokollon keresztül.

Eredetileg Unixra írták, de ma már futtatható VAX VMS, Novell NetWare, OS/2 és AmigaDOS operációs rendszereken is.

Mielőtt nekifognánk a SaMBA részletes ismertetésének, néhány fontosabb fogalom:

- **Share:** megosztott erőforrás. Ez lehet file vagy nyomtató szolgáltatás. A megnevezése \\server\share módon történhet.

- **Browsing:** a gépek és gépeken lévő share-ek listájának összeszedése, megfelelő hirdetése, keresés a gépek vagy share-ek között.

- **Workgroup:** egy munkacsoportba tartozó gép. Minden workgroupnak van egy master browser. A gépek szavazással döntenek el, hogy ki legyen a master browser.

- **Domain:** tartomány. Ez nagyon hasonlít a workgroupra, de annál legalább egy dologban többet tud: tartalmaz egy kitétetett számítógépet, a tartományvezérlőt (domain controller).

- **Domain controller:** tartományvezérlő gép, tulajdonképpen a szervernek megfelelő. Ha a kliensek számára előírjuk egy tartományba való belépést, akkor nem a kliens, hanem a tartományvezérlő végzi el a felhasználónév/jelszó ellenőrzést, sikeres belepés esetén pedig lefuttathat egy logon scriptet a klienssel.

## A SaMBA fő részei

**smbd:** Az SMB-szerver. Ez kezeli a kapcsolatokat, ellenőrzi a jogokat és végzi az aktuális adatátvitelt.

**nmb:** Netbios Name Server. Segít a klienseknek megtalálni a szervereket, ő intézi a böngészést (browse), és a maga módján a tartományvezérlő (domain controller) szerepét is betöltheti.

**smbclient:** Unixos kliens program, ftp-szerű felhasználói illesztővel.

**smbn:** Egy kis „glue” (ragasztó) program, hogy az smbd biztonságosan tudjon külső programokat hívni.

**testprns:** Program a szerver nyomtatóeléréseinek tesztelésére.

**testparms:** Program a konfigurációs file szintaktikus ellenőrzésére.

**smb.conf:** A SaMBA konfigurációs file-ja. Általában a /usr/local/samba/lib/smb.conf helyen található, de ha Linux disztribúció (mint például a Debian) tartalmazza, akkor az /etc/smb.conf helyre kerül.

**Dokumentáció:** Nagyon hasznos. Nemcsak a SaMBA-hoz, hanem a kereskedelemben kapható egyéb SMB-s rendszerek belső lelkivilágáról is nagyon érdekes dolgokat tudhatunk meg :-). Ha mi telepítjük a SaMBA-t, akkor a /usr/local/src/samba-1.9.16p11/docs/ könyvtárban található, a disztribúció részeként pedig az usr/doc/samba he-

## A SHARE-EK ÉS A GROUPOK JOGAI

jogok/felhasználók:	user	jht	pusr	tdk	efi	guest
homes	-rw	-rw	-rw	-rw	-rw	---
printers	x---	xw-	x---	xw-	x---	x---
admin	x---	xrw	x---	x---	x---	x---
tdk	x---	x---	x---	xrw	x---	x---
alkalmazások	x---	xr-	xrw	xr-	x---	x---
anonftp	xr-	xr-	xr-	xr-	xr-	xr-
netlogon	xr-	xr-	xrw	xr-	xr-	xr-
efi	---	---	---	---	-rw	---

A fenti táblázatban az *r* olvasási (read), a *w* írási (write) az *x* böngészési (browse) jogot jelent. Egy sor egy share, egy oszlop egy group.

lyen keresendő. Továbbá vannak szabványos Unix-kézikönyvlapok, nagyon tanulságos a *man smb.conf*.

## Elterjedtebb SaMBa-kliensek

A SaMBa-szerverek nemcsak Linux alól érhetőek el, hanem szinte minden PC-s operációs rendszer alól be tudunk alájuk jelentkezni:

- LAN manager client for DOS: Ez egy szabadon letölthető Microsoft-kliens. Előnye: DOS alatt működik. Hátrányai: max. 8 karakter hosszú share nevek, sok konvencionális RAM-ot fogyaszt.

- Windows for Workgroups 3.11: Nem ingyenes, de a TCP/IP stack szabadon letölthető hozzá. A Trumpet Winsock nem elég, mert nem csak az alkalmazásoknak, de a rendszernek is kell a TCP/IP. Nemcsak kliens, hanem szerver is lehet. Az angol verzió 8. lemezen található az ADMINCFG.EXE, ezzel le lehet tiltani a password cache-t, ami elég nagy biztonsági rést üthet az egyébként jól védett Unixos rendszerünkön is. (A password cache file-ok felhasználónév.pwl nevűek, és többek között tartalmazzák azt, hogy az adott felhasználó melyik share-re, milyen jelszóval csatlakozott. Elvileg biztonságos a kódolása, de tudott, hogy melyik pozíció tartalmazza a felhasználónevet – ami a file neve is!-, ezt kihasználva egy jó gépen néhány óra alatt feltérhető). A Microsoft TCP/IP stack nehezen él együtt az IPX/SPX-szel, ha ilyenre van szükség, akkor inkább használjunk win95-öt.

- Windows 95: Egyes verzióknál, ha megosztott nyomtató van, akkor a \WINDOWS\SYSTEM\könyvtár mindenki által olvasható. A rendszer itt

tárolja a \USERNEV.PWL file-okat. Lehetséges következményei már ismertek fentebbről.

- Windows NT 3.x WorkStation: Megbízható kliens, de sajnos a SaMBa még nem tud domain controller lenni NT-knek. Bővebb információt erről a /usr/doc/samba/DOMAIN\_CONTROL.txt file-ban olvashatunk.

- Windows NT 4.x WorkStation: Kedveli a titkosított jelszavakat. Ha egyszer egy gépről megjegyzi, hogy tudja kezelni a titkosított jelszavakat, akkor később már szóba sem áll vele, ha az a gép nem így kéri. Ha ilyen kliensünk vagy szerverünk van, akkor jobb a legutolsó SaMBa verziót használni. Ez most a SaMBa 1.9.16p11, de párhuzamosan készül a jelentősen átdolgozott 2.0 verzió is.

Néhány kliens nem rendelkezik olyan lehetőséggel, hogy más felhasználóként kapcsolódjunk egy share-hez, mint ahogy bejelentkeztünk. Ez a „Kapcsolódás másként” (Connect As). Ekkor használhatjuk a \\server\share%username formát. Ez akkor is használható, ha azt tapasztaljuk, hogy a kliens figyelmen kívül hagyja azt, amit megadunk neki a fenti dialogusablakban.

Miután végigfutottunk néhány alapfogalmon, nézzünk meg egy SaMBa-konfigurációt! Példakonfigurációként bemutatom egy általam évek óta használt rendszer beállításait:

## A felhasználók jogai

Egy felhasználó vagy a *user*, vagy a *jht* csoportba tartozik elsődlegesen. A többi csoport úgynevezett másodlagos. (Egy már meglévő felhasználót az „adduser lajbi másik” paranccsal tehetünk be másodlagos csoportba.



Az 1. táblázatban egy lehetséges csoport-jogosultság hozzárendelés látható. Szavakkal kifejtve: a *user* csoport tagjai elérhetik saját *home* könyvtárukat írásra-olvasásra, és az *anonymous ftp* területet olvasásra. A *jht* csoportnak van egy közös munkaterülete *admin* névvel. A *pusr* (*power user*) csoport írhat is az alkalmazások *share*-re. Ők installálhatnak a kliensek számára programokat. A *tdk* csoport hasonló a *jht*-hoz, de ők nem az *admin* területét használják, hanem annak egy alkönyvtárát, *tdk* névvel.

Az *efi* *share* érdemel egy kis figyelmet. Ez senki számára nem *browse*-olható, tehát csak úgy lehet elérni, ha valaki tudja, hogy létezik, és beírja, hogy „\\jht\efi”. Innentől már a normális kapcsolódási eljárás folyik le. A fentieket megvalósító *smb.conf* konfigurációs file az 1. listában látható.

A *SamBa* a *valid users* opcióval szabályozza, hogy ki kapcsolódhat az adott *share*-hez. Az elindított *smbd* process ugyanolyan jogokkal rendelkezik, mint a belépett felhasználó, vagy – ha van – amit a *force user* és *force group* előír. De ez még nem jelenti azt, hogy minden file fölött uralkodik, mert a hagyományos *unixos* jogosultságok érvényben vannak. Ez nem egyszerű, mert három szinten szabályozhatjuk a hozzáféréseket:

1. Ki kapcsolódhat az adott *share*-hez? (*valid users*)
2. Ha kapcsolódott, akkor milyen jogosultságúvá válik? (*force user*, *force group*)

### TOVÁBBI SAMBA-INFORMÁCIÓK

A cikk időnként frissített verziója megtalálható a <http://jht.gau.hu/~lajbi/sambaoladas.html> címen.

A *SamBa* Web-lapjai: <http://lake.canberra.edu.au/pub/samba/>, ahol lehet keresni a levelezési lista archivumában is.

A *SamBa* levelezési lista (angol): [samba@samba.anu.edu.au](mailto:samba@samba.anu.edu.au) A *SamBa* újdonságai (angol): [samba-announce@samba.anu.edu.au](mailto:samba-announce@samba.anu.edu.au).

A beiratkozás ezekre úgy történhet, hogy küldünk egy levelet a [listproc@samba.anu.edu.au](mailto:listproc@samba.anu.edu.au) címre. A Subject legyen üres, és a levél törzsében: *subscribe samba* (nevünk) vagy *subscribe samba-announce* (nevünk) szöveg szükséges.

Elég sokszor beszélnek a *SamBáról* a [comp.protocols.smb.newsgroupban](http://comp.protocols.smb.newsgroupban) is.

3. A *unixos* file-ok hagyományos hozzáférés szabályozása (tulajdonos, csoport, többiek).

Ezért nézzünk még két konfigurációs file-t. A 2. listában egy rövid */etc/passwd* részletet, a 3. listában egy */etc/group* részletet láthatunk.

## Logon scriptek

Mivel a *linuxos* gép tartományvezérlő is, szükség van a kliensek számára lefutatható *logon* scriptre is. Mivel az *smb.conf*-ban a „*logon script = %G.bat*” sor szerepel, a beléptetett felhasználó elsődleges csoport neve szerinti *batch* file fog elindulni. Ezek:

*users.bat*:

```
net use i: \\jht\homes /yes
net use h: \\jht\tdk /yes
net use w: \\jht\alkalmazasok /yes
```

*jht.bat*:

```
net use i: \\jht\homes /yes
net use h: \\jht\admin /yes
net use w: \\jht\alkalmazasok /yes
```

## Az *smb.conf* security opciója

Az *smb.conf* *security*-ja három lehetséges értékkel bírhat:

- *share*: Ez akkor jó, ha többnyire olyan felhasználók kapcsolódnak a *SamBához*, akiknek nincs *account*-juk a *unixos* gépen.

- *user*: Ez akkor javasolt, ha a klienseken és a *Unixon* a *login* nevek megegyeznek. Gondot okozhat néhány *furcsa* tulajdonsággal rendelkező kliens néham. A *wfmg* például ebben az esetben néha teljesen figyelmen kívül hagyja a „kapszolódás másként” *dialogus*dobozba irt nevet és jelszót.

Úgy tűnik, némelyik *NT 4-es*nél van hasonló *mizéria*, ekkor használjuk a *\\server\share%user* formát. Bár ezt meg nem minden kliens fogadja el :-).

- *server*: Ekkor meg kell adni a *password server = gepnev* opciót is, és a felhasználói név/jelszó ellenőrzést az adott gép fogja elvégezni. Ez általában egy *NT AS* szokott lenni.

Hogyan is működik mindez? Az *SMB*-szerver a kapcsolat létrehozásakor közli a klienssel, hogy milyen üzemmódban van. Ez azt befolyásolja, hogy a kliens milyen módon próbálja azonosítani magát:

**security = user**: Először erről, ez az egyszerűbb. Ez esetben a kliens küld egy *session setup* blok-

## 1. LISTA: A CSOPORTOK JOGAIT LEÍRÓ SMB.CONF FILE

```

; Configuration file for jht.
;
=====
; For the format of this file and comprehensive
; descriptions of all the
; configuration option, please refer to the man page
for smb.conf(5).
;
; last update: Lajbi 96.11.05
[global]
    server string = %h Jarmu- es Hotechnika tanszek, \
Samba %v
    workgroup = JARMUTECHNIKA
    printing = bsd
    printcap name = /etc/printcap
    load printers = yes
    guest account = guest
    security = user
    hosts allow = 192.188.244. 192.188.243.
192.188.245.
    domain logons = yes
    domain master = yes
    guest ok = no
    logon script = %G.bat
    max disk size = 120
    log file = /usr/local/samba/var/log.%m
    lock directory = /usr/local/samba/var/locks
    share modes = yes
    os level = 34
    remote announce = 192.188.243.255/JHT
192.188.245.255/JHT
    message command = /usr/bin/mail -s 'message from \
%f on %m' root < %s; rm %s
; saját home könyvtár mindenkinek
[homes]
    comment = Home directories
    browseable = no
    read only = no
    create mode = 0700
    mangled map = (*.html *.htm)
; nyomtatás csak a jht és tdk csoportnak
[printers]
    comment = All Printers
    path = /var/tmp
    browseable = no
    printable = yes
    public = no
    writable = no
    create mode = 0700
; magánterület
    valid users = %jht, %tdk
; közös munkaterület a jht group tagjainak, full
; access minden tagnak
[admin]
    comment = Tanszek kozos dolgai
    path = /home/admin
    valid users = %jht
; ók kapcsolódhatnak
    public = no
    writable = yes
    printable = no
    force user = admin
    force group = jht
; a létehozott file-ok tulajdonosa az admin.jht lesz
    create mode = 770
; a csoport szabadon garázdálkodhat
; közös munkaterület a tdk group tagjainak, az [admin]
részé!
[tdk]
    comment = TDK-s hallgatók terulete
    path = /home/admin/tdk
    valid users = %tdk
    public = no
    writable = yes
    printable = no
    force user = admin
    force group = jht
    create mode = 770
; DOS/Windows binárisok a jht, tdk groupnak és guest
; usernek, de read-only
; A pusr group írhat rá!
[dosbin]
    comment = DOS es Windows alkalmazások
    path = /home/jht/dosbin
    writable = no
    printable = no
    write list = %pusr
    valid users = %jht,%tdk
    force user = jht
    force group = jht
; ez a dosbin még egyszer
[alkalmazások]

```

```

copy = dosbin
;annon ftp terület, de itt read-only az incoming is
[anonftp]
comment = JHT anonymous ftp terulete
path = /home/ftp/pub
public = yes
writable = no
printable = no
; domain controller miatt
[netlogon]
comment = Domain Controller logon directory
path = /home/jht/netlogon
writable = no
printable = no
guest ok = yes
write list = @pusr
browseable = no
force user = jht
force group = jht
; ezek itt nem browse-olhatók, csak azoknak, akik
; tudnak róla
; egy project által használt terület
; efi: EFI project tagjainak (efi group)
[efi]
comment = EFI project
path = /home/efiproject
valid users = @efi
public = no
writable = yes
printable = no
force user = lajbi
force group = efi
create mode = 770
browseable = no

```

ket közvetlenül a protokoll egyeztetése után. Ez tartalmaz egy felhasználói név/jelszó párost. Ez esetben a szerver még nem tudja, hogy ez a kliens milyen share-ekhez fog fordulni, tehát a megadott felhasználói név, jelszó és természetesen a gép ismeretében dönthet arról, hogy elfogadja-e vagy nem a kérést.

Ha ez megtörtént, akkor a kliens elvárja, hogy a későbbiekben a tree connection blokkal mountolni tudjon share-eket.

Egy kliens több session setup kérést is küldhet. A szerver a válaszban ad egy „felhasználóazonosítót”, így a kliens elvileg képes többszörözött kapcsolatok fentartására.

**security = share:** Ez az alapértelmezett a SaMBA-ban, mert ez volt készen elsőnek. A kliens ekkor azonosítani próbálja magát minden tree connect esetében, vagyis amikor mountolni akar egy share-t. Ekkor viszont nem küld explicit felhasználói nevet, csak a kért share nevét és egy jelszót! A kliensek ekkor azt várják, hogy egy jelszó egy share-hez tartozik, és nem egy felhasználóhoz. A Windows for Workgroups alatt és a Windows 95-ben nincs felhasználói azonosítás, így arra nem lehetett építeni a védelmet, ezért ott egy share-hez tartozik egy jelszó. A SaMBA mindig a Unix rendszerű felhasználó/jelszó séma alapján azonosít és dönt a hozzáférésről.

Sok kliens küld session setup blokkot még ak-

kor is, ha share-szintű az ellenőrzés, és ebben megad egy felhasználói nevet, de nem ad jelszót. A SaMBA feljegyzi ezt a nevet egy lehetséges felhasználók listára. Amikor a kliens küld egy tree connection blokkot, akkor az ebben kért share neve is felkerül a lehetséges felhasználók listájára, továbbá az összes, az smb.conf file adott share-hez tartozó valid users sorában szereplő nevek.

Ha a küldött jelszó bármelyik, a lehetséges felhasználók listáján szereplő névhez illeszkedik, akkor a kliens számára a talált névhez tartozó jogokkal engedélyezett a hozzáférés.

**security = server:** Ebben az esetben a SaMBA azt mondja a kliensének, hogy *security = user-on* van. Amikor megkapja a felhasználói név/jelszó párost, akkor ezeket használva a SaMBA megpróbál belépni a password serverre. Ha ez sikerül, akkor elfogadja a kérést, különben nem.

A SaMBA a kapcsolat létrehozásakor nem csak azt közli, hogy milyen üzemmódban van, hanem azt is, hogy képes-e titkosított jelszavakat kezelni. Ha ezen opcióval fordítottuk a SaMBA-t, akkor visszaküld a kliensnek egy kulcsot is. Ez esetben a kliens az összes jelszavát titkosított formában fogja küldeni.

Ekkor külön jelszó file-t kell karbantartani a SaMBA számára, ugyanis kriptográfiailag lehetetlen a Unix típusú jelszavakat SaMBA típusúra konvertálni.

## 2. LISTA: EGY RÖVID /ETC/PASSWD RÉSZLET

```
lajbi:xxx:510:101:Lajber Zoltan,aula 120,1535,,,:/home/lajbi:/bin/bash
znagy:xxx:505:100:Nagy Zoltan:/home/znagy:/bin/bash
czanik:xx:512:100:Czanik Andras,ELTE JTK,,,:/home/czanik:/bin/bash
admin:xxx:1002:101:Tanszekei adminisztracio,aula 125,1430:/home/admin:/bin/bash
jht:xxxxx:1003:101:./home/jht:/bin/bash
fjuhasz:x:1008:100:Juhasz Ferenc,GM IV,,,:/home/fjuhasz:/bin/bash
jritz:xxx:1013:100:Ritz Jozsef,GM V/8,C/4026,,,:/home/jritz:/bin/bash
guest:xxx:1016:109:./home/guest:/bin/true
```

## 3. LISTA: EGY /ETC/GROUP RÉSZLET

```
users:*:100:
jht::101:lajbi,ari,jbeke,llaib,admin
pusr::103:jht,lajbi
tdk::104:znagy,tracz,losgyan,fjuhasz,horvath,jritz,bhamza,beno
efi::107:lajbi,fjuhasz
```

Előnyként jelentkezik, hogy bizonyos kliensek felhasználói távolból is megváltoztathatják a jelszavukat, ehhez nem szükséges bejelentkezniük pl. telnettel a SaMBA-s gépre.

## A SaMBA mint faxszerver

A forrás: /usr/doc/samba/Faxing.txt-ben, a be szerzési helyekkel együtt. A faxszerverhez az aláb-  
biakra van szükség: UNIX SaMBA-val, Linux pre-  
ferált, Ghostscript, mgetty+sendfax csomag és  
pbm csomag.

Fax küldéséhez az alábbiakat kell tennünk:

- A SaMBA-val felállítunk egy postscript nyom-  
tatót, ami fizikailag nem létezik. Ehhez kapcsolód-  
hatnak a kliensek, és nyomtathatnak is rá.

- Megírjuk a faxot, amiben csak akarjuk. A lé-  
nyeg az, hogy a fejlécben a címzett fax száma  
előtt mindig ugyanaz a jól azonosítható string áll-  
jon, pl. Faxszám.

- Küldjük el a faxot az ályomtatóra, így az be-  
kerül a spoolba, postscript formátumban. A  
postscript alkalmas arra, hogy az ismert string  
utáni részből kiszedjük a faxszámot, ahová küldeni  
akarjuk a faxot. Ez a fő trükk.

- Ez után a pbmtext programmal előállítjuk  
a fax fejlécét (az apróbetűs részt), majd a  
postscript doksit átalakítjuk Ghostscripttel G3

fax formátumúra, és az mgetty segítségével el-  
küldjük.

Az érkező faxok fogadása sem ördögösség: az  
mgetty fogadja is a faxot, és azokat át tudjuk ala-  
kítani pl. a g3tolj vagy hasonló eszközökkel. Ezt  
követően küldhetjük nyomtatóra, vagy pl. ghost-  
view-val képernyőre. Vagy küldhetünk e-mailt, pl.  
az „smbclient -M jucika < fax jött!” paranccsal ér-  
tesítve a felhasználót.

## Egyéb szolgáltatások

**DHCP:** Ez hasonlít a bootp-hez, de elvileg többet  
tud nála. A megfelelő bootpd-vel lehet DHCP-t is  
szolgáltatni, de csak statikusan, nincs lejáratí ide-  
je az IP-címeknek. A Debian 1.2-ben:

```
/usr/sbin/bootpd -v
bootpd+dhcp 2.4.3
```

Ekkor ez alkalmas rá. Nem kell semmi különös  
beállítás, a hagyományos /etc/bootptab használ-  
ható, sőt még a bf=netzlmage is maradhat a disk-  
less Linuxokhoz. Ha nem ilyen a bootpd, akkor  
upgrade-re van szükség, vagy marad patch és az  
újrafordítás.

**WINS:** Windows Name Server, a DNS és a net-  
bios nevek közötti megfeleltetést segíti, alapértel-  
mezten be van kapcsolva. A Debán van NTAS  
WINS a hálózaton, ki kell kapcsolni.

## Összehasonlítás

Mint minden rendszer, a SaMBA sem univerzális. Lássunk néhány érvet és ellenérvet a SaMBA mellett és ellen:

Amit a SaMBA NEM tud (biztos van más is, de amivel én találkoztam eddig):

- A Windows NT-k titkosítási algoritmusai nem ismert, így a SaMBA még nem tud együttműködni ezekkel sem Primary, sem Secondary Domain Controllerként, vagy logon serverként.

- Nem lehet központi helyről RAS-sal karbantartani a felhasználók gépeit.

Amit a SaMBA tud, de a WinNT Server nem (csak néhány dolog, ami eddig feltűnt nekem):

- NT-n nem ismerek módszert a host allow = és host deny = típusú korlátozásra. Ez akkor jelent gondot a WinNT-nél, ha van TCP/IP a NetBEUI mellett, és így az egész világ láthatja szerverünket.

- Távoli propagálás lehetősége. Ezzel elérhető, hogy a szerverünk olyan hálózatokon is feltűnjön, ahol a normál browsing nem működne. Egy példa: remote announce = 192.168.2.255/SERVERS 192.168.4.255/STAFF. Ezzel elérhető, hogy a 192.168.2.0-s hálón mint SERVER, míg a 192.168.4.0-s hálózatban mint STAFF feltűnjön gépünk.

Érvek a Linux + SaMBA ellen:

- Nem tud tartományvezérlő lenni Windows NT-k számára.

- Nem tud távoli elérést (RAS).

- Közösen írható/olvasható területeken a file lockolás viszonylag lassú lehet, ha az alkalmazások sok apró file-t használnak. Ez ellen segít, ha a csak olvasható share-eket „share modes = no”-ra állítjuk, vagy a SaMBA-t megfelelő opciókkal újrafordítjuk. Lásd: /usr/doc/samba/Speed.txt.gz.

Érvek a Linux + SaMBA mellett:

- Nagyon jó a teljesítménye. Többnyire 30%-kal jobb, mint más PC-s szerverprogramoké, az átviteli sebesség megközelíti az 10 Mbytes ethernet elvi sebességét. Saját méréseim szerint jellemző a 900-1000 Kbyte/s-os olvasási sebesség.

- Kicsi az erőforrás-igénye. Az első linuxos, SaMBA-s szerverem hardverfejlesztés nélkül megy évek óta mint tanzéki szerver. Ez a gép 15 db Windows for WorkGroups 3.11 részére tartományvezérlő, innen futtatják a Microsoft Office-t, Matlab, Simulink és egyéb alkalmazásokat, POP3 segítségével leveleznek. Emellett további 30 felhasználó használja telnettel, e-mail szerver, Web-szerver és egy levelezési lista is üzemel a

gépén. A gép konfigurációja: OPTI485SLC VLB alaplap, AMD486DX40 CPU, 16 Mbyte RAM, VLB-s IDE vezérlő. Ma már az a furcsa helyzet állt elő, hogy szinte az összes kliens gép ennél erősebb.

- Nyílt rendszer. Ez Linux-tulajdonság is, nem csak a SaMBA-é. A Linuxra fel lehet kapcsolni NFS (UNIX) NCPFS (Novell NetWare) és SMBFS (Win85, WinNT, LANMan) köteteket, és ezeket tovább is tudja adni. Ez roppant kényelmessé teszi például a biztonsági mentéseket különböző platformok esetén is.

- Nagyszámú felhasználó esetén is könnyen adminisztrálható, hiszen nemcsak grafikus felületen keresztül, hanem unixos parancssorból, illetve parancsfile-okból is kezelhető, ami hozzáértő kezeken rendkívüli hatékonyságot eredményez.

Sok sikert a linuxos és bármilyen más SaMBA-záshoz! ■

## Linux Station

A CHIP-CD-n kipróbálhatja, nálunk megveheti az eredetit!

Applicware for Red Hat Linux, Dev./Edu.	99,800 / 18,800
Caldera Open Linux 1.1, Base/Standard	17,800 / 88,800
Caldera Wabi 2.2 for Linux	49,800
Doctor Linux, 5/E v. Linux Bible, 4/E w/CD	8,800
GNUstep for Linux	6,800
Linux Developer's Kit - 11 CD Set!	6,800
Linux Dev's Resource/Linux Toolbox (InfoMagic)	5,800 / 8,800
Linux Man Pages, 2/E (Red Hat)	8,800
Moo-Tiff for Linux (InfoMagic)	26,800
Red Hat Linux 4.2 for Intel / Alpha v. Sparc	8,800 / 10,800
Red Hat Linux Library CD, Vol. 3	5,800
Red Hat Linux Power Tools (6 CD Set)	4,800
Red Hat's Motif for Linux (Book + CD)	35,800
Red Hat's Tritreal CDE v1.2 - Client / Dev.	18,800 / 46,800
Slackware Linux 3.3 (!) - Walnut Creek	4,800
Unixfix Linux System v2.0 (InfoMagic)	21,800

Angol nyelvű szakkönyvek legnagyobb választéka!



1111 Bp. Karinthy F. 25.

☎ 371-0704  
209-5951

<http://www.swsbooks.hu>

Czakó Krisztián  
(slapic@fido.hu)

# Linux az irodában

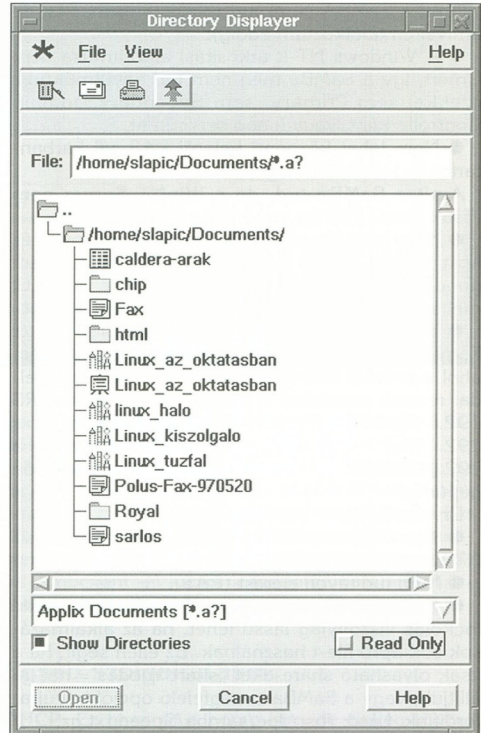
A számítógépek egyik leggyakoribb felhasználási helye az iroda, általában szövegszerkesztőre, táblázatkezelőre, a cég adatait tároló adatbázisra, néha számlázásra és mostanában egyre inkább elektronikus levelezésre használják a gépet. Valamivel ritkábban, de megvan az igény grafikák, bemutatók készítésére, Web-lapok tervezésére is.

A következőkben szeretnénk bemutatni (a teljesség igénye nélkül) a fenti feladatok megoldására néhány lehetőséget a Linux segítségével.

A legtöbb feladatra általában találunk szabad szoftvereket, bár ezek sosem annyira komplexek, csicsásak, mint a bolti társaik, viszont a célok 90%-ának megvalósítására alkalmasak. Egy másik hátrányuk, hogy nem tartalmaznak filtereket, konvertereket más szoftverekkel készült anyagok beolvasásához, azok formátumában történő mentéshez, illetve az ő formátumaikat nem ismerik a bolti szoftverek. (Ami nem mindig ezen programok hibája, gondoljunk a TeX-re, mely egy igen szabványos és professzionális szövegformázó rendszer.) A szabad szoftverek által használt formátumok is, ellentétben sok bolti szoftverrel, teljesen szabadok és ismertek. Mint azt fent említettem, szép számban vannak bolti programok is Linuxhoz, melyek egyrészt adatformátumukban kompatibilisek más operációs rendszerekkel elérhető irodai programokkal, másrészt azokhoz igen hasonló felépítéssel, kezelőfelülettel rendelkeznek, mely egyszerűvé teszi az átállást rájuk. Itt most két irodai programcsomagot és egy önálló szövegszerkesztőt kívánunk ezek közül bemutatni.

## Applixware

Az Applixware egy eredetileg is Unix platformra készült irodai rendszer. Jelenlegi szolgáltatásai közé tartozik a szövegszerkesztő, táblázatkezelő, rajzoló-



és bemutatókészítő, adatbázis-kezelő, e-mail- és alkalmazásfejlesztő. A jelenlegi linuxos verzióban az adatbázis-kezelő kivételével (igazság szerint az adatbázis kapcsolati rendszer, mely ODBC adatbázisokhoz képes kapcsolódni nincs benne, de a szabványos fejlesztői felületnek köszönhetően egyedül esetre elkészíthető) minden benne van. Jelenlegi verziója, a 4.3-as április elején jelent meg, és kétféle csomagban létezik. Az ún. *Office Suite* csomagban az alkalmazásfejlesztőt leszámítva minden megtalálható, és ötvenezer forint körüli ára igen kedvező.

Az alkalmazásfejlesztőt is tartalmazó *Develop-*

ment *Kit* csomag ennél borsosabb, százezer forint körüli áron kapható.

A kiváló minőségű alkalmazásfejlesztő rendszerének köszönhetően könnyedén írható hozzá mindenféle kiegészítés, például cégadatbázis-kezelés. Jelenleg magyar változatban egy általános cégadatbázis rendszer létezik, mely leltározásra, ajánlatkészítésre, számlázásra, ügyfényilvántartásra van felkészítve.

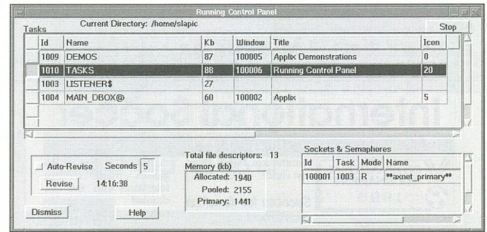
A rendszer honosítása folyamatban van, a menü-rendszere már teljesen magyar, a program többi üzenetének fordítása ősz végére fog elkészülni. Magyar elválasztó- és helyesírás-ellenőrző modul is része lesz a csomagnak, amelyek a Morphologic Helyes-e? és Helyesel programjainak változatai lesznek.

Számos, más szoftverek által készített állománytípus képes beolvasni, illetve ezekbe exportálni, filterek segítségével. Nagy előnye, hogy teljesen objektumorientált, minden belső funkció elérhető a makrónyelvén keresztül is, amellyel könnyedén lehet komoly formázó-, rajzolóprogramokat készíteni. A saját dokumentumai között, valamint külső dokumentumból kapcsolatokat lehet létrehozni, azaz például egy táblázatkezelőben készült táblázatot be lehet illeszteni egy szövegbe, amit a rendszer automatikusan frissít, ha a táblázatkezelőben módosítjuk az adatokat. Ugyanez működik a grafikákkal és az összes általa ismert dokumentumformátummal is.

Az e-mail rendszerrel a dokumentumokat azonnal elküldhetjük másoknak, amit a program automatikusan tud konvertálni a címjegyzékben beállított formátumra. Ugyanez működik a nekünk küldött dokumentumok beolvasásakor is. Egy egyszerű fax makróval könnyedén küldhetünk faxokat is, illetve a Linux rendszer által fogadott faxokat is azonnal megnézhetjük, ha szükséges nyomtathatjuk.

A program erőforrás-igénye alkalmazási környezettől függően változik. Kényelmes használatához legalább 16, de inkább 32 Mbyte RAM ajánlott. Viszont a memóriagény nem nő számottevően az elindított alkalmazások számának növekedésével, mivel a teljes rendszer osztott eljáráskönyvtárakkal működik. Ez azt jelenti, hogy egy szerver gépen, 32 Mbyte RAM-mal akár egyszerre négyen-öten is dolgozhatnak (ennek tényleges megvalósításához az X-Window rendszer ad tökéletes lehetőséget).

A szövegszerkesztő és a táblázatkezelő a fenti extra szolgáltatásokon kívül a megszokott funkciókat adja, valamint képes HTML file-ok szerkesztésére is. A grafikus programész egy ilyen komplex programcsomaghoz képest igen komoly szolgálta-



tásokkal rendelkezik (vektoros és pixeles szerkesztés, szöveg görbére illesztése, képelemek stb.). Az e-mail egy viszonylag egyszerű, de minden szükséges szolgáltatást tartalmazó elem, míg a programfejlesztő környezete igen jó, grafikus ablaktervezővel, debuggerrel felszerelve.

Összességében egy kitűnően használható, szinte korlátlan fejlesztési lehetőségekkel felszerelt rendszer. Érdemes megjegyezni, hogy az Applix cég elkészítette a program Java-verzióját Applix Anywhere néven. (A demó változat a <http://www.applix.com/> címről indulva megtekinthető.)

## StarOffice

A StarOffice 3.1 a Microsoft Office csomagjához igen hasonló, szintén komplett irodai alkalmazás-csomag. A program a német StarDivision cég terméke, és – a cég adatai szerint – mostanra elhódította a német piac 90%-át, ami a Microsoftra szemben nem kis siker. A Linux-változat ingyenesen letölthető az Internetről, és saját célra szabadon használható. A céges változatát a Caldera forgalmazza.

Az újabb, 4.0-s verzióknak már javas változata is van, így ismét egy platformfüggetlen szoftvercsomaghoz juthatunk.

A programcsomag szövegszerkesztőt, táblázatkezelőt, rajzolóprogramot és bemutatókészítőt tartalmaz, a 4.0-s verzióba pedig adatbázis-kezelőt is beépítettek (dBase formátumú és ODBC adatbázis-kezeléssel). A programcsomag egyik nagy előnye, hogy otthoni használatra és oktatási intézmények számára ingyenes. Az üzleti Linux-változat ára ötvenezer forint körül van, a Caldera önállóan is forgalmazza, illetve része a Caldera OpenLinux összes verziójának. Magyar honosítása még ennek sem létezik, esetleges jövőbeli elkészítéséről sincs még hírvünk.

Közvetlenül képes kezelni sok más programcsomag által használt adatformátumot, és ezt igen jó

**International Soccer**  
REPORT '96

months are being given first choice.

**Success for American Tournament**  
The overwhelming success of the International Tournament boded well for the future of the games later in the year. Each

than 45,000. One of the games made national television, with the U.S. dramatically coming back from a

three goal deficit to lose only by one goal in the last moments of the game.

**Tips Sheet**  
The Angle play drill was designed to force the goalkeeper to make a quick decision on whether a ball will go in or out of the goal without interference. The goalkeeper

PER GAME TICKET PRICES			
Competition	Call	Call2	Call3
Opening Game	\$120	\$70	\$40
Game One	\$75	\$45	\$30
Round One	\$65	\$45	\$35
Round Two	\$65	\$40	\$35
Quarterfinals	\$140	\$80	\$55
Semifinals	\$300	\$200	\$90
Third-Place Game	\$135	\$100	\$60
Final	\$425	\$300	\$180

FAI 07/06/96: 100% of 100%

színvonalon teszi. A ma már kihagyhatatlannak számító HTML szerkesztési lehetőséget is tartalmazza, a szokásos szövegszerkesztői színvonalon.

Erforrás-igénye igen magas, amit még a külön pénztér megvásárolható Motif igény is tetéz. Természetesen ez a program is osztott eljáráskönyvtárat használ, ami az Applixware-nél leirtakhoz hasonló lehetőségeket nyújt. (Ez általában így van Linux-programok esetében.) Jelenlegi verziója ugyan e-mail rendszert nem tartalmaz, de a dokumentumainkat el tudja küldeni elektronikus levélben. Szövegszerkesztője igen sok mintadokumentumot tartalmaz például faxok elkészítéséhez, de készíthetünk vele saját mintákat is, melyeket később felhasználhatunk. Ehhez menüvezérelt segédrendszer ad kitűnő támogatást. A programcsomag a szokásos, minden irodai csomagban megtalálható funkciókkal rendelkezik. A StarOffice egy igen komplex, sok csicsát és extra funkciót tartalmazó, egyébként elég általános irodai alkalmazáscsomag. Előnyei: több platformon is (Linux, Windows 3.1x, Windows 95) gyakorlatilag meg egyező külsővel elérhető és nagymértékben kompatibilis más rendszerekkel.

## WordPerfect

A WordPerfect 7.0 a Corel cég ismert szövegszerkesztője. Linux alatt még csak béta változatban létezik, de már igen stabil és jól használható rend-

szer. A program a szövegszerkesztő mellett tartalmaz egy egyszerű rajzolóprogramot és táblázat-szerkesztőt is. Utóbbival diagramokat is készíthetünk.

A WordPerfect is tartalmaz adatbázis-csatlakozási lehetőséget ODBC rendszerrel. Jelenleg az Oracle-t kezelő kapcsolat készült el, de a dokumentáció ígér további adatbázis-kapcsolatokat is. Mivel végleges verzió még nincs, az áráról sincsenek még információk. A jelenleg kapható korábbi – 6.0-s – változat ára ötvenezer forint körül van.

Magyar változatról nincs hír, bár a konfiguráció felajánlja a választható nyelvek között a magyart is, de sem az elválasztórendszer, sem a helyesírás-ellenőrző nem működik ezzel a beállítással. Minden biztonnyal ugyanaz a file kell hozzá, mint a más platformokon kapható változathoz, így azokkal együtt várhatóan ehhez is lesznek magyar kiegészítések.

Az elengedhetetlen adatfile-kompatibilitás itt is megvan, képes kezelni más rendszerek formátumait. Itt is megtalálható a HTML-szerkesztés, de sokkal összetettebb formában, mint a többi programnál. Egy külön almenü van erre a célra, és igen jó minőségben konvertálja saját és más programok dokumentumait HTML formába, beleértve a grafikákat is, melyekre a linkek is jól helyezték a HTML file-ban. Az eddig kipróbált szövegszerkesztők közül messze a WordPerfect készíti a legjobb HTML-dokumentumokat. (A „natív” HTML-szerkesztőkhöz hasonlítani kár lenne, bár egész jól közelíti azok képességeit.)

Erforrás-igénye viszonylag alacsony, működési sebessége kifejezetten jó. Kinézetre nem hasonlít a többi szövegszerkesztőre, a szokásos Corel féle külön jegyeket hordozza, ami munka közben sokszor alkalmasabbnak bizonyult a csicsás, „mindenhol ikonok” megoldással szemben. Nagyon kellemes szolgáltatása a környezetérzékeny menürendszer, mellyel az adott szöveg vagy grafikai részhez használható menüket azonnal, a bekezdésnél bárhol kattintva előhozhatjuk.

Aki már dolgozott a WordPerfect régebbi verzióival, akár DOS alatt, akár grafikus rendszerekben, és megszerette azt, itt sem fog csalódní, a programot azonnal ki tudja használni. A program a fentiek kívül tartalmaz egy egyszerű e-mail részt, mellyel nemcsak a dokumentumunk küldhető el másoknak, hanem rövid leveleket is írhatunk.

Mivel alapvetően csak egy szövegszerkesztő, nehéz összehasonlítani a többi programmal. Kiegészítések nélkül (táblázatkezelő, e-mail, grafikai program, prezentáció) sajnos nem elég egy iroda számítógépesítésére. ■

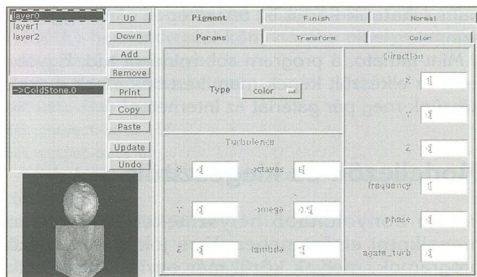


Cseke Balázs  
(sinus@freeseide  
.elte.hu)

# Trace-eljünk Linux alatt!

Raytracing, magyarul fénykövetés. Ez az a módszer, amivel élethű, fotorealisztikus képeket lehet számítógépen generálni. Gondoljunk csak mondjuk a Jurassic Park dinóira vagy a Toy Storyra.

Bizonyára sokan szeretnének hasonló képeket (animációkat) készíteni. Ez nem olyan nehéz, hiszen itt nem is számít annyit a rajztudás, mint mondjuk a hagyományos kétdimenziós képek készítésénél. Eddig PC-n főleg DOS/Windows alá léteztek ilyen programok. Amikor (még az 1.2.x sorozat idején) „rátaláltam” a Linuxra, még csak ismerkedtem a raytrace technikával. Akkortájt kezdtem el használni a PoVray (Persistence of Vision) nevezetű programot, még DOS alatt. Azt azonban nagyon untam, hogy a hosszadalmas számolások alatt nem tudok mit kezdeni a géppel, így egy időre pihentettem a dolgot. Amikor feltelapítottam a Linuxot, megtaláltam a PoVray linuxos verzióját is. Gondoltam, legalább megnézem,



A POVTextureEditor

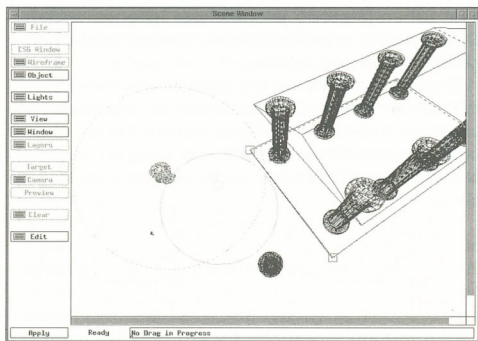
mit bír egy ilyen unixos multitask. Kicsit leest az állam, amikor két PoVray futott egyszerre, és még olvasgatni is tudtam mellette a gépen! Azóta sok idő eltelt, a Linux is, a PoVray is sokat fejlődött, és megjelent számos segédprogram is.

Persze más raytrace program is létezik Linux alá: Polyray, BMRT, MTV, Radiance és más kisebb csomagok. Most a PoVrayt mutatom be, mégpedig azért, mert két ponton is hasonlít a Linuxhoz: ingyenes és fantasztikus dolgokat tud.

A PoVray és az előbb felsorolt programok egy ponton mind egyeznek: szövegfájlban kell megadni a kiszámítandó képet. Ez első hallásra ijesztő, de egyrészt később lesz szó modellező programokról, másrészt egy idő után ennek inkább az előnyeit érzi az ember.

## A PoVray eszköztára

Mit kínál a PoVray? Számos alaptestet: kocka, kúp, gömb, henger, tórusz, melyeket CSG (Constructive Solid Geometry) segítségével összeadhatjuk, kivonhatjuk egymásból, így természetesen új testeket hozhatunk létre. Képet domborzattá alakíthatunk úgy, hogy a világosabb pontok lesznek magasabban. A tárgyra számos előre elkészített textúrát feszíthetünk, de a defini-

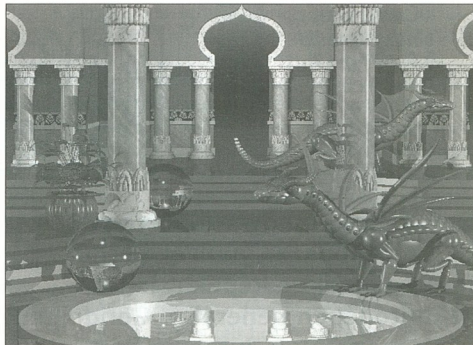


A SCED modellező működés közben

álás olyan tág teret ad, amelynek csak a fantáziánk szab határt.

A fényfajtából is több van: pontfény, testfény (területen világít, így sokkal lágyabb árnyékokat ad), fényszóró. Ezeket az atmoszféra beállításával láthatóvá lehet tenni, akár csak a levegőben táncoló porszemeket. Az aura (haló) képes létrehozni a Nap vagy a Hold fényudvarát, lézert, robbanást, lángot. Külön létezik még szivárvány is. Alkalmazhatunk ködöt is, tetszőleges sűrűséggel és színnel.

Mint látható, a program sok mindent tud. Egyébként az elkészült képek magukért beszélnek – csak nézzünk meg pár galériát az Interneten!



## Modellezők és kiegészítők

Persze bonyolultabb helyszíneket szövegesen megalkotni elsöre nehéz. Erre jök a modellező programok, melyek közül már egyre több jelenik meg Linuxra. A következőkben a legelterjedtebből láthatunk néhányat.

A *SCED* az én kedvencem. Sok funkció még hiányzik, de már most jól használható. Legnagyobb előnye, hogy a CSG műveletek eredményét kiszámolja és úgy jeleníti meg. Nagyon pontosan lehet vele tárgyakat elhelyezni, ugyanis egy ún. „constrain” rendszert használ. Ez azt jelenti, hogy a tárgyakat mindig valamilyen kényszerítő erővel mozgatjuk: pl. egy körön, vagy két sík metszésvonalán, vagy egy gömb és egy egyenes közös pontjain (az ugye kettő darab). Érdemes a programmal elkészíteni a helyszínt, majd kézzel beírni a textúrákat és a speciális effektusokat. A program ingyenes.

A *SCED*-hez jó kiegészítő az *AC3D*, bár teljesen más jellegű. Itt nem magukat a tárgyakat, hanem azok alkotópontjait lehet manipulálni. Tud DXF-et, Lightwave állományt olvasni, és PoVray-t, VRML-t írni. A *SCED*-hez jó kiegészítő, ha az általa generált PoV file-okat befűzzük a kész *SCED* file-ba. E program shareware (20\$).

Egész más a *Midnight Modeller*: ezt DOS-ból hozták át, és ez sajnos meg is látszik rajta. Egyébként elég sok mindent tud (pl.: forgástegek, speciális tárgyak, jól konfigurálható külső), de sajnos a CSG műveleteket nem számolja ki, hanem az alaptestekkel jeleníti meg. Ha valakit nem zavar a 8+3 betűs file-név, akkor használható, csak vigyázni kell testek közötti kivonásnál a sorrendre. Egyébként ez is ingyenes.

És végül a *Leo3D*: egészen friss, új modellező. Nagyon szép és kényelmes kezelői felülettel. Egyelőre CSG-t nem tud, de logó készítéséhez ideális. Viszont shareware (25\$).

A modellezőprogramokkal még nem merült ki a PoVray-ezni vágyók eszköztára. Itt van a *POV-TexureEditor*. Ez egy Java alatt futó textúraszerkesztő. Nagyon kényelmes, az egyik legfontosabb kiegészítő PoVray-hez (legalábbis szerintem). Legnagyobb előnye, hogy látjuk a textúrát készítés közben. Kell hozzá a JDK 1.0.2. és természetesen ingyenes.

Aki X-Window-t használ, annak javaslom a *TkPoVo-t*. Ez egy grafikus indítófelület. Szép, kényelmes és gyors. Beállítható tetszőleges szerkesztőprogram is. Ha már segédprogramokról van szó, bizonyos funkciókat a PoVray saját leíró nyelvével is el lehet látni. Jó példa erre a „Lens Flare” hatás. Ez a kamera becsillanását utánozza, illet őrben játszódo jeleneteknél lehet sokat látni. De fellelhető egy olyan file is, amivel élethű fákat generálhatunk képeinkre.

Érdemes felrakni a *xfractintet* is, az ezzel generált képek kitűnőek domborzatnak. Számos más program is jó ha kéznél van: ilyen a *Gimp*, amivel utólag csinosíthatjuk képünket, vagy az *xv*, amivel különböző formátumokba menthetjük a kész alkotást.

Akit komolyabban érdekel a dolog, az feltétlenül nézzon el a <http://www.povray.org> honlapra. Magyarul a Generations oldalain (<http://www.datanet.hu/generations>) készül a kezdőknek egy hosszabb bevezetés ebbe a csodálatos világba.

Mint látható, eszközök és segítség van, a többi már csak kedv, fantázia és idő kérdése. ■

Vörösbaranyi  
Zoltán  
(vbzoli@hbrt.hu)

# Merre, hogyan magyarul?

Gyakran felvetődő kérdés: ha már van egy működő Linux rendszerünk, miként tudunk azon magyarul dolgozni, ékezetes betűket írni-nyomtatni. A másik gyakori probléma az, hogy hol lehetünk magyar nyelvű Linux-dokumentációra, illetve hol kérhetünk és kaphatunk anyanyelvünkön segítséget.

Az Interneten magyarul fellelhető egy tipp-trükk gyűjtemény e témában, a *Magyarul-HOGYAN?*, ennek lapunk készítésekor legutolsó (1997. május 3-i 0.17-es) verziója olvasható alább.

Természetesen a *Magyarul-HOGYAN?* folyamatosan megújul, az újabb változatok az <ftp://ftp.tarki.hu/pub/magyar/linux/doc/Magyarul-HOGYAN>, vagy a <http://www.tarki.hu/Linux/Magyarul-HOGYAN.html> címről tölthetők le.

A *Magyarul-HOGYAN?*-t Vörösbaranyi Zoltán szerkeszti, az alábbi lelkes társaság közreműködésével:

Fülöp Szilárd (fulopsz@edasz.hu)  
Magossányi Árpád (mag@bunuel.tii.mata.vu.hu)  
Markó István (mi@bgtyf.hu)  
Soós Péter (sp@osb.hu)

Szakacsits Szabolcs (szaka@mirror.cc.u-szeged.hu)

Külön köszönet illeti Soós Pétert, aki az utóbbi időben a fő közreműködője a doksi fejlesztésének. Természetesen sokat köszönhet a *Magyarul-HOGYAN?* a többi nemzeti HOWTO szerzőinek, a Linux fejlesztőinek és másoknak, akiket most kifelejtettünk.

Ezek után lássuk, hogyan tehetjük komfortossá Linuxunkat!

## A bash és a tcsh magyarítása

Gyakori probléma az, miként lehet rávenni a bash-t és a tcsh-t, hogy magyar karaktereket is elfogadjon?

A következő sorokat kell beírni a felhasználó saját (HOME) bejelentkezési könyvtárába elhelyezett `.in` putrcr nevű állományba:

```
bash használata esetén:
set meta-flag on
set convert-meta off
set output-meta on
tcsh használata esetén (/etc/csh.login, vagy
.tcshrc) a
setenv LC_CTYPE ISO-8859-1
stty pass8
sorok vezetnek eredményre.
```

## A less magyarítása

A less esetében hasonlóan egyszerű a megoldás, a `LESSCHARSET=latin1` export `LESSCHARSET` sorokat kell behelyezni a `/etc/profile` állományba. (Vagy a `.profile`-ba).

## Latin-2 (ISO-8859-2) kódkiosztás konzolon

A Unix világában elterjedt 8-bites kódkiosztás, amely tartalmazza a magyar ékezetes betűket is az ISO-8859-2, azaz a Latin-2 kiosztás. Ez a kódkiosztás tartalmazza a latin betűs szláv nyelvek (horvát, szlovén, szlovák, cseh, lengyel), valamint a magyar, román és német nyelv ékezetes betűit.

A nyugat-európai országok az ISO-8859-1 kódkiosztást használják (Latin-1). A Latin-1 kiosztás tartalmazza a magyar ékezetes betűket is (ugyanazon kóddal) az ő és ű kivételével. Az ő és ű betűk helyén a Latin-1-es kiosztásban az õ és ű szerepel; így a Latin-2-ben kódolt magyar szövegek elolvashatók Latin-1-es kiosztás használatával is (jobb híján).

A Linux rendszermag alapesetben a Latin-1-es kiosztást használja úgy, hogy a Latin-1-es kódo-

kat leképezi a PC-s 437-es kódlapra. Ezzel a módszerrel csak olyan Latin-1 betűket tud megjeleníteni, melyek szerepelnek a 437-es lapon. A Latin-2-es kódkiosztás használatához két dolog szükséges:

- a karakterkészlet betöltése (VGA/EGA monitorok esetén),

- valamint a fent említett leképezés átállítása normál leképezésre (betűkód = képernyőkód).

A leképezés megváltoztatását a következő parancs végzi el:

```
echo -ne '\033(K'
```

A *mapscrm* parancsral lehet egyébként a leképezést átállítani. A Latin-2-es karakterkészlet betöltése (slackware linux esetén):

```
setfont /usr/lib/kbd/consolefonts/lat2-16.psf
```

A fenti két sort célszerűen a */etc/rc.d/rc.local* állományba kell bemásolni.

Latin-2-es betűkészletek beszerezhetők az Internetről, a

- <http://goliat.eik.bme.hu/~bartoki/linux/font/sd.latin2.tar.gz>,

- <ftp://ftp.tarki.hu/pub/magyar/linux/console/lat2fonts.tar.gz>

- és <ftp://ftp.tarki.hu/pub/magyar/linux/console/sd.latin2.tar.gz> címekről. (Ez utóbbi csomag egy jó leírást tartalmaz a linuxos konzol-fontokról is, szerzője Bartók István.)

A */usr/lib/kbd/consolefonts* könyvtárba kell másolni az állományokat.

## Latin-2 kiosztás használata X-Window felületen

Az X-Window grafikus felület is a Latin-1-es kódkiosztást használja alap esetben. Latin-2 betűket a következő helyeken találni:

- <ftp://ftp.tarki.hu/pub/magyar/X11/fonts/>, valamint a

- <http://sizif.mf.uni-lj.si/linux/cee/iso8859-2.html> címen is érdemes körülnézni.

## Nyomtassunk!

Hogyan nyomtassunk Latin-2 kódolású text file-okat bármilyen – Ghostscript által támogatott, vagy postscript – nyomtatón?

(A megoldás nem Linux-specifikus, elvileg bármilyen Unix alatt működik.)

Először postscript formátumra kell hozni a szöveget. Erre számos program létezik, ezek közül a GNU *enscript* ismeri a Latin-2 kódrendszert. Be kell tehát szereznünk a GNU *enscript* programot, továbbá mivel az Adobe fontjai nem tartalmazzák a Latin-2 karaktereket, egy olyan fontot, amely tartalmazza őket. Ilyen pl. az IBM Courier, amely benne van az XFree86 disztribúcióban: a file-ok neve cour.afm és cour.pfa, courb.afm és courb.pfa stb.

Telepítsük az *enscript* programot, majd helyezzük el egy alkönyvtárba a cour\*.afm és cour\*.pfa file-okat, majd adjuk ki a következő parancsot ebben az alkönyvtárban:

```
mkafmmap *.afm
```

Ezután az *enscript* tudomására kell hozni, hogy létezik a fenti alkönyvtár. Ehhez módosítanunk kell az *enscript* lokális vagy globális konfigurációs állományában az *AFMPath* opciót, valahogy így:

```
AFMPath: /usr/local/lib/psfonts/ibmcour:/usr/local/lib/enscript
```

Természetesen a fenti elérési útvonalak attól függenek, hogy hová telepítettük az *enscript*-et, illetve az IBM Courier fontokat. Ezek után egy *enscript -xlatin2 latin2\_kodolasu\_szoveg\_file.txt* parancsnak ki kell nyomtatnia az alperjelmezett nyomtatón a „latin2\_kodolasu\_szoveg\_file.txt” file-t a megfelelő Latin-2 karakterekkel.

Használhatjuk filterként is az *enscript*-et. Ilyenkor az adott nyomtató filtere valahogy így néz(het) ki:

```
#!/bin/sh
DEVICE=bj200
RESOLUTION=360x360
PAPERSIZE=a4
SENDEOF=
ENCODING=latin2
enscript -M4 -XENCODING -Z -B -g -p- |
if [ "$DEVICE" = "PostScript" ]; then
    cat -
else
gs -q -sDEVICE=$DEVICE \
-r$RESOLUTION \
-sPAPERSIZE=$PAPERSIZE \
-dNOPAUSE \
-dSAFER \
-sOutputFile=- -
fi
if [ "$SENDEOF" != "" ]; then
    printf "\004"
fi
```

Részletesebb információk az *enscript* és a Ghostscript dokumentációjában találhatóak.

## Magyar billentyűzet konzolon és X-Window alatt

A Linux konzolon a *loadkeys* parancs használható a billentyűzetkiosztás beállítására; paraméterként egy definíciós állományt vár, melyben tetszőleges kódot rendelhetünk a billentyűkhöz. A *loadkeys* programot a rendszer indításakor érdemes futtatni. A Slackware kiadásban a */etc/rc.d/rc.local* állományba érdemes tenni (más kiadásban esetleg a */etc/rc-be*, vagy a */etc/rc.local-ba*):

```
/usr/bin/loadkeys /usr/lib/kbd/keytables/magyar.map.
```

A következő kiosztások (vagy csomagokban található kiosztások) használhatók a fenti *magyar.map* helyén (ezeket kell bemásolni a */usr/lib/kbd/keytables/* könyvtárba):

- *ftp://ftp.tarki.hu/pub/magyar/linux/console/magyar.map*: szabványos magyar kiosztás,
- *ftp://ftp.tarki.hu/pub/magyar/linux/console/hun-on\_us.map* (lásd az *ftp://ftp.tarki.hu/pub/magyar/linux/console/README.Linux* leírást),
- *ftp://ftp.tarki.hu/pub/magyar/linux/console/humap.tgz*.

Természetesen X-Window alatt is írhatunk magyarul, a billentyűzetet tetszés szerint át lehet definiálni akármilyen magyar kiosztás megvalósítására, akár átválthatóra is. A billentyűzet kiosztása az *xmodmap* programmal történik, amely bemenetül egy szöveges definíciós állományt vár. Ezt bárkikor lefuttathatjuk, de a rendszer indulásakor a *HOME/.Xmodmap*-ból, vagy ha ez nem létezik, akkor a */usr/X11R6/lib/X11/xinit.Xmodmap*-ból állítódik be a kiosztás.

Ilyen definíciós állományokat a következő helyeken találhatunk:

- *ftp://ftp.tarki.hu/pub/magyar/X11/keyboard/xmodmap.hu*: szabványos magyar billentyűzethez,
- *ftp://ftp.tarki.hu/pub/magyar/X11/keyboard/xmodmap.std.hu*: szabványos magyar billentyűzethez, repülőékezet-támogatással (a szerző Soós Péter),
- *ftp://ftp.tarki.hu/pub/magyar/X11/keyboard/xmodmap*: átváltható kiosztás.

Az *xmodmap* állományt *.Xmodmap* néven a *HOME* könyvtárba, vagy a */usr/X11R6/lib/X11/xinit* könyvtárba kell másolni. Az első esetben az adott felhasználónál érvényesül a kód kiosztás, míg a másodikban mindenkinél (l. a */usr/X11R6/lib/X11/xinit/xinitrc-t*).

Ha a fenti szabványos *xmodmap*-ot (*xmodmap.hu*) kívánjuk használni, akkor az *XF86Config* file-nak a

```
# To set the LeftAlt to Meta, RightAlt key to ModeShift,
# RightCtl key to Compose, and ScrollLock key to
ModeLock:
```

```
# LeftAlt      Meta
# RightAlt     ModeShift
# RightCtl     Compose
# ScrollLock   ModeLock
```

részében a „*# RightAlt ModeShift*” sor elől el kell távolítani a *#* jelet, engedélyezve ezzel az *AltGr* funkciót.

## Magyar Slackware csomag

A *hufont-0.2.tgz* csomag a Slackware Linux kiadás támogatására készült. Megtalálható az alábbi helyen:

```
ftp://ftp.tarki.hu/pub/magyar/linux/console/hufont-0.2.tgz.
```

Részlet a *README*-ből:

1. Ez Slackware package. Más disztribúciókból indítsd el a */etc/rc.d/rc.font*-ot valahonnan (pl.: */etc/rc.d/rc.local*).

2. Lehet, hogy a *tic* nincs a *PATH*-ban, akkor azt a sort csinálj meg kézzel a */install/doinst.sh*-ből.

3. Lesz egy új termináldefiníció, *console-lat2* néven. Ez a *lat2-\*.psf* fontokkal megy, az *iso02.\*-gal* nem. Nem tudom, melyik a szabvány.

4. A */etc/skel*-ben van *.inputrc*, meg *.joerc*, de nincs *.bashrc* vagy ilyesmi. Az alábbi sorokat érdemes beletenni:

```
export LESSCHARSET=latin1
if [ $TERM="console" -o $TERM="linux" ]
then
```

```
    export TERM="console-lat2"
```

```
fi
```

## Hol található magyar kiegészítések Linuxhoz?

Még egyszer, összefoglalva – magyar billentyűzetkezelők és betűkészletek Linux konzolhoz:

- *ftp://ftp.tarki.hu/pub/magyar/linux/console/*,
- *http://www.tarki.hu/ftp/pub/magyar/linux/console/*.

*X11*-hez:

- *ftp://ftp.tarki.hu/pub/magyar/X11/*,
- *http://www.tarki.hu/ftp/pub/magyar/X11/*.

## TeX

Létezik egy magyar elválasztóprogram TeX/LaTeX alá, a HiOn, vigyázat, a program nem szabad szoftver. Soós Péter kiegészítése ezzel kapcsolatban:

„Átnéztem a HiOn (így kell írni) doksját, és meglepve tapasztaltam, hogy a HiOn nemcsak egy elválasztóprogram a TeX-hez, hanem egyben egy kódkonverziós program is. Képes mindenféle kódolásra (CWI, 852, Latin-2, 123) mindenféle konvertálni (CWI, Latin-2, 123, TeX) szövegfile-okat.”

Szerinte a következő paranccsal lehet Linux alá lefordítani:

```
„cc -DF_INDEX_DEFINED -o hion hion.c”.
```

A HiOn a <http://WWW.iSYS.hu/c/verhas/hion/index.html> címen érhető el.

## LyX

Magyar karaktereket sokféleképpen elő lehet csalogtatni a LyX-ből. Az egyik lehetséges megoldás (felteve, hogy magyar nyelvű kiegészítést használunk):

Állítsuk be a lyxrc (.lyxrc vagy system.lyxrc) file-ban a következőt:

```
\fontencoding T1.
```

Ezek után, ha a Layout → Document → Encoding értéket Latin-2-re állítjuk, vagy a lyxrc-be beírjuk, hogy

```
\inputencoding latin2,
```

akkor a LyX helyesen fogja nyomtatni a magyar ékezetes karaktereket. Célzerű Latin-2 kiosztású képernyőfontokat használni a fenti esetben, hogy már ott is jól jelenjenek meg a magyar betűk. Ehhez be kell állítani az Options → Screen Fonts párbeszédpanelben a megfelelő fontokat, és a Font Norm értékét iso8859-2-re állítani, vagy a lyxrc-be beírni a

```
\roman_font
```

```
\sans_font
```

```
\typewriter_font
```

sorokba a megfelelő fontok neveit, és a következő sort:

```
\font_norm iso8859-2.
```

Latin-2 fontok találhatóak a következő helyeken:

- <ftp://ftp.tarki.hu/pub/magyar/X11/fonts/>,

- <ftp://ftp.osb.hu/pub/misc/fonts/local/>.

Célzerű még a magyar helyesírásnak megfelelő idézőjelekhez a Layout → Quotes párbeszédpanelben a idézőjelek típusát *German Double* értékre állítani, vagy a lyxrc file-ban a következő sorokat beírni:

```
\quotes_language german
```

```
\quotes_times 2.
```

Célzerű még a Layout → Document → Language értékét „magyar”-ra állítani, vagy a lyxrc-be a

```
\language magyar
```

sort beírni. Ez engedélyez bizonyos, a magyar nyelvre jellemző beállításokat (pl. dátumformátum) a TeX számára. A magyar nyelv implementálása sajnos még nem teljes a TeX rendszerben. Ha magyar elválasztást is akarunk, akkor be kell szereznünk a TeX Babel csomagjához egy magyar elválasztás file-t amely elérhető a következő címen:

- <ftp://ftp.cs.elte.hu/pub/hzoli/private/hungarian-latex.tar.gz>.

A fentieket a LyX 0.10.7 és a TeX TeX disztribúció 0.3.4 változatával teszteltem. (Soós Péter)

## A joe „magyarítása”

A joe editornak a .joerc ill /usr/lib/joe/joerc file-jában be lehet állítani a -asis opciót, illetve ^T és Meta-chars-as-is menüpont alatt lehet futás közben megadni, hogy rendesen kijelezze az ékezetes betűket. Ez csak az újabb joe-val megy így, de szerintem nagyon ritka az olyan Linux, ahol régi joe van. (Magossányi Árpád)

## Emacs ékezetekkel

Az Emacs ékezetes használatához a .emacs, vagy a /usr/lib/emacs/site-lisp/default.el konfigurációs állományt az alább látható sorokkal kell kiegészíteni: (standard-display-european t)

```
(set-input-mode (car (current-input-mode))
                (nth 1 (current-input-mode))
                0)
```

További hasznos sorok a .emacs file-ban:

- (*require 'iso-syntax*): Az iso-syntax segítségével helyesen fog működni a kis- és nagybetűk közötti konverzió.

- (*require 'iso-transl*): Az iso-transl lehetővé teszi az ékezetes betűk beírását olyan terminálokon is, amelyek nem képesek ékezetes betűkódokat generálni. Használat:

```
Ctrl-x 8 ' a → á
```

```
Ctrl-x 8 " o → ö
```

```
Ctrl-x 8 ~ o → ő
```

```
Ctrl-x 8 ^ u → ü
```

```
stb...
```

– Az *iso-accents-mode* is hasonló funkciót lát el, de használatakor elhagyhatjuk a C-x 8 prefix-et, valamint ez a mód mindig csak az adott bufferre aktivizálódik. Az *M-x iso-accents-mode* paranccsal kapcsolhatjuk be. (Markó István)

További információ az Emacs info file „European Display” szekciójában. Hasznos lehet még az isohun package is, amellyel az adott bufferben található repülő ékezetes szöveget konvertálhatjuk ISO-8859-re. Ez letölthető az <ftp://ftp.bgytf.hu/pub/emacs/misc-packages/iso-hun.el> címről.

- <ftp://ftp.osb.hu/pub/linux/>: RedHat RPMS, Netscape.

- <ftp://fang.fa.gau.hu/pub/>: Java, Java-dokumentációk, SUSE, Spinner...

- <ftp://ftp.jate.u-szeged.hu/pub/linux/>: RedHat, Slackware, friss kernelék, Linux Gazette, Netscape, Wingz, WINE, HTML-szerkesztők...

- <ftp://ftp.cs.elte.hu/pub/>: zsh, teTeX, SPARC-Linux, ELM, XFree86.

- <ftp://xenia.sote.hu/pub/linux/>: doksik, hálózati dolgok, programfejlesztési eszközök...

## A magyarítás témájához kapcsolódó leírások

Érdemes elolvasni a linuxos konzol-fontok és unicode iránt érdeklődőknek Bartók István ([bartoki@goliat.eik.bme.hu](mailto:bartoki@goliat.eik.bme.hu)) kiváló leírását, amelyet a következő helyen találhat az érdeklődő:

- <ftp://ftp.tarki.hu/pub/magyar/linux/console/sd.latin2.tar.gz>

- <ftp://ftp.osb.hu/pub/linux/magyar/fonts/console/sd.latin2.tar.gz>.

A fenti állományból a README tartalma: kis Linux font & console OLVASSEL, FAQ, Q&A.

## Magyar vonatkozású linuxos ftp-helyek

- <ftp://ftp.kfki.hu/pub/linux/>: a sunsite.unc.edu tükörhelye.

- <ftp://sunsite.math.klte.hu/pub/linux/>: a sunsite.unc.edu/pub/Linux tükörhelye.

- <ftp://ftp.bme.hu/pub/OS/Linux/>: a tsx-11.mit.edu tükörhelye.

- <ftp://ftp.bme.hu/pub/gnu/>: GNU programok.

- <ftp://ftp.vein.hu/linux/>: X11, apps, doc, kernel, packages, system, utils, xsources...

- <ftp://ftp.math.bme.hu/pub/>: XFONMS, fonts, hungry\_programers, LyX, Netscape, math, samba, ssa, tex, gap...

- <ftp://ftp.math.bme.hu/pub/linux/>: RedHat, Slackware, új kernelék, doksik, dosemu.

- <ftp://ftp.iit.uni-miskolc.hu/pub/>: Slackware, RedHat, Apache...

- <ftp://ftp.tarki.hu/pub/magyar/linux/>: linuxos magyar kiegészítések, betűkészletek, billentyűkezelők, doksik.



## Magyar WWW-helyek

- <http://www.cab.u-szeged.hu/local/linux/>: A Magyar Linux Felhasználók Címlapja (Karbantartó: Szakacsits Szabolcs [szaka@mirror.cc.u-szeged.hu](mailto:szaka@mirror.cc.u-szeged.hu)).

A következő helyeken tükrözik ezeket a lapokat:

- <http://www.sienet.hu/linux/> (Nagykanizsa),

- <http://fang.fa.gau.hu/mlf/> (Gödöllő),

- <http://bucipc.math.klte.hu/dlug/mlf/index.html> (Debrecen),

- <http://www.westel.hu/mlf/> (Budapest),

- <http://www.cis.ufl.edu/~vs/linux/> (Gainesville, FL, USA),

- <http://www.agria.datanet.hu/mlf/> (Eger),

- <http://pirix.idg.hu/mlf/> (Budapest),

- <http://www.dunanet.hu/mlf/> (Dunaújváros),

- <http://www.Hnet.isys.hu/mlf/> (Gyöngyös),

- <http://www.szentes.datanet.hu/mlf/> (Szentes),

- <http://www.datanet.hu/generations/linux/hunindex.html>: Generations, Amit a Linuxról tudni szeretne (Karbantartó: Nyitrai Tamás [defiant@mail.datanet.hu](mailto:defiant@mail.datanet.hu));

- <http://mla.telnet.hu/>: A Magyar Linux Alapítvány címlapja (Karbantartó: [john@graphisoft.hu](mailto:john@graphisoft.hu));

- <http://www.kfki.hu/linux-int/>: a Linux International tükörzése;

- <http://www.vma.bme.hu/linuxdoc/linux.html>: The Linux Documentation Project Home Page;

- <http://xenia.sote.hu/hu/sysadm/#linux>: Linux HOWTO doksik;
- <http://bucipc.math.klte.hu/dlug/>: Debreceni Linux-felhasználók oldala;
- <http://esca.atomki.hu/dlug/>: Linux Felhasználók Köre MTA ATOMKI, Debrecen;
- <http://www.idg.hu/internetto/stb/linux/>: Linux Fórum az iNteRNeTTo (Internet-újság) hasábjain;
- <http://www.kfki.hu/~kadlec/Mirror.html>: Linux/Unix anyagok KFKI-s tükrözése;
- <http://www.avia.kfkipark.hu/list4/>: a Linux-kernel levelezési lista archívuma;
- <http://www.avia.kfkipark.hu/list2/>: a Linux-admin levelezési lista archívuma;
- <http://info.fsz.bme.hu/brokers/ftp/query.html>: magyarországi ftp-helyekhez kereső.

## Magyar levelezőlisták

- **Linux-kezdő lista:**  
Csatlakozás: a [linux-kezdoo@garfield.sch.bme.hu](mailto:linux-kezdoo@garfield.sch.bme.hu) címre a következő tartalmú levelet kell írni: „subscribe linux-kezdoo”.  
A lista címe: [linux-kezdoo@garfield.sch.bme.hu](mailto:linux-kezdoo@garfield.sch.bme.hu).
- **Magyar Linux-felhasználók listája:**  
Csatlakozás: a [linux-request@cc.u-szeged.hu](mailto:linux-request@cc.u-szeged.hu) címre a következő tartalmú levelet kell írni: „subscribe vezetéknev keresztnév”.  
A lista címe: [linux@cc.u-szeged.hu](mailto:linux@cc.u-szeged.hu).  
A <http://www.vma.bme.hu/~tiv/lla/> címen a lista archivált változatát találhatjuk meg.
- **Linux-dokumentációk fordítóinak listája:**  
Csatlakozás: a [listserv@tohotom.vein.hu](mailto:listserv@tohotom.vein.hu) címre a

következő tartalmú levelet kell írni: „Subscribe linux-doc Vezeteknev Keresztnév”.

A lista címe: [linux-doc@tohotom.vein.hu](mailto:linux-doc@tohotom.vein.hu).

● **Linux-vitalista:**

Csatlakozás: a [linux-flame-request@csomalungma.csoma.elte.hu](mailto:linux-flame-request@csomalungma.csoma.elte.hu) címre a következő tartalmú levelet kell írni: „subscribe”.

A lista címe: [linux-flame@csomalungma.csoma.elte.hu](mailto:linux-flame@csomalungma.csoma.elte.hu).

● **Szabad Szoftver Alapítvány listája:**

Csatlakozás: a [listserv@tohotom.vein.hu](mailto:listserv@tohotom.vein.hu) címre a következő tartalmú levelet kell írni: „Subscribe ssa Vezeteknev Keresztnév”.

A lista címe: [ssa@tohotom.vein.hu](mailto:ssa@tohotom.vein.hu).

## Újságok

- **hun.lists.mlf.linux:** a Magyar Linux-felhasználók listájának újságváltozata;
- **hun.lists.mlf.linux-kezdoo:** A Linux-kezdő lista újságváltozata;
- **hun.lists.mlf.linux-hw:** A Linux-hardware listának újságváltozata;
- **hun.lists.mlf.linux-flame:** A Linux-flame listának újságváltozata;
- **hun.lists.mlf.linux-doc:** Linux-dokumentációk fordítói listájának újságváltozata;
- **hun.lists.mlf.unix:** a Unix-levelezőlista újságváltozata;
- **hun.comp.lang.java:** A Java programnyelvről;
- **hun.comp:** a számítástechnikáról általában;
- **hun.lists.hix.guru:** a számítástechnikáról általában. ■



Négyesi Károly  
([chx@cs.elte.hu](mailto:chx@cs.elte.hu))

# File-rendszerek

**A Unix-alkalmazások sebessége – azonos processzorú és kiépítettségű gépeken – jelentősen meghaladhatja DOS-os és windowsos rokonaikét. Az eltérést nem csak a 16 bitesnél nagyobb (32 és 64 bites) címzési mód és adatszélesség kihasználása magyarázza: a lemezműveletek nagyobb hatékonysága is hozzáadódik ehhez.**

Egy Linux rendszerben többfajta file-rendszer közül választhatunk, érdemes tehát elgondolkoznunk azon, melyik mire képes, és melyikre van leginkább szükségünk. Többfelhasználós rendszerről lévén szó, tudni kell, hogy kié a file. Ez a Linux rendszerben a felhasználói azonosító (UID) és a csoportazonosító (GID) tárolását jelenti. Szintén szükséges a különböző csoportok jogainak tárolása. Ebben az esetben ez a tulajdonos, az ő csoportja, és mindenki más különböző jogait jelenti. Szükség lehet még az utolsó hozzáférés és az utolsó módosítás idejének tárolására. Az sem meglepő, hogy hosszú file-neveket is szeretnénk használni. Bár nem természetes kívánság, de gyakran szükséges a file endjeleiknek utolsó változtatási idejét is eltárolni.

## Lemez, partíció, file-rendszer

Tisztázni kell egy fogalmat, ami a legtöbb – DOS, Windows alatt dolgozó – emberek ismeretlen lehet: ez a file-rendszer. E fogalmon egy operációs rendszer által használt file-tárolás struktúráját és módszereit, más szavakkal a file-ok rendszerezését értjük. Ez DOS alatt a sokszor elátkozott FAT-ra épül. Unix alatt rengeteg különféle file-rendszert használhatunk egy rendszer alatt. (DOS alatt is vannak más file-rendszerek: a különféle röptömörítők tömörített file-rendszert valósítanak meg az eredeti FAT rendszerre ráépülve, illetve a Windows 95 a VFAT vagy FAT32 rendszert használja, ami kevéssel különbözik a FAT-tól.)

Mi a különbség egy partíció vagy akár az egész diszk és a file-rendszer között? Ha valaki a partícióhoz nyúl, akkor csak a nyers szektorok egymásutánját éri el. Ha

a file-okkal dolgozik, akkor a file-rendszert kell elérnie a rendszer szolgáltatásaival. A DOS egy nagy hibája, hogy ezt a két fogalmat összemossa, lehetetlenné téve ezzel a biztonságos tárolást. (Az INT 25h/26h az adott partíción belül címzi a lemezt, lehetőséget nyújtva ezzel sok programnak a file-rendszer megkerülésére.)

Ezek után lássuk a Linux által augusztus elején támogatott főbb file-rendszereket! Nem véletlen ez a fogalmazás: hetenként megjelenik egy-két új kernelverzió, és bármelyikben lehet egy új file-rendszer.

## A Minix és leszármazottai

Az ős Linux file-rendszere Andrew Tannenbaum Minix operációs rendszerének file-rendszere volt. Linus Torvalds ezt használta a Linux rendszer kifejlesztésénél. Pontosabban a Minix file-rendszert, annak kiosztását vette át, de a kódot ő írta meg hozzá. Ez nem meglepő, ha tudjuk, hogy Linus egy 40 Mbyte-os vinyóval megáldott gépen kezdte fejleszteni az operációs rendszert, és egész egyszerűen nem volt hely még egy partícióra és azzal együtt egy másik file-rendszerre... Ennek a file-rendszernek *Minix-1* a neve, hogy megkülönböztesse a többi Minix-verziótól. Ez a file-rendszer a legstabilabb, mivel a legrégebbi és a legjobban tesztelt. Nem véletlen, hogy a bootlemezek többségén Minix file-rendszert találunk. Erre magyarázat még az is, hogy ez a rendszer elég gyors is. Csak egyetlen időmezőt támogat, a file utolsó módosításának az idejét. (Ezen kívül még kettőt szokás tárolni: az utolsó hozzáférés és az inode utolsó módosításának idejét.) Sajnos csak 14 karakteres file-neveket és legfeljebb 64 Mbyte-os összméretet enged meg. A Linux operációs rendszer egyetlen dolgot adott az eredeti Minix file-rendszerhez: a szimbolikus linkeket. Ez annyit jelent, hogy egy file-ra több könyvtárból, több néven is hivatkozhatunk. A *minix-30* kiterjesztette az előző rendszer képességeit azzal, hogy 30 karakteres neveket tud használni. Érdekesképpen megjegyezzük, hogy a kernelben azonos kód vezényli a két rendszert, de sok program – például maga a Minix operációs rendszer – miatt muszáj kétféle rendszernek tekintenünk. A *xafs-t* Frank Xia készítette. Ez nem más, mint a *minix-1* 248 karakterre bővítése. E file-rendszer nagy előnye, hogy az egyik leggyorsabb rendszer (valószínűleg a leggyorsabb, de ezt nagyon nehéz eldönteni). Hátránya, hogy hiányzik belőle a két időmező.

## A kiterjesztett rendszerek

A következő file-rendszer már Remy Card fejlesztése. Ez az extended (kiterjesztett) file-rendszer az *extfs*. Ez Linus minix-1 rendszerére épül, de már nincs sok köze hozzá. 255 karakteres file-neveket, hatalmas partíciókat támogat, és a hiányzó két időmezőnek is jutott benne hely. Egyetlen „apró” hibája a lassúság. Lassabb mint akár a minix, akár az *ext2fs*, akár a *xiafs*.

Természetes, hogy ezt a sebességproblémát meg akarták oldani. Szintén Remy Card volt, aki újraírta. Ez lett a második kiterjesztett file-rendszer, az *ext2fs*. Ez már elég gyors, mára már eléggé stabilis is vált. Egyedülálló képességei, sebessége, és kitűnő tervezése – amely bőven hagy helyet a növekedésre – mára a legelterjedtebb linuxos file-rendszerré tették.

## ext2fs kontra xiafs

Az *ext2fs* és a *xiafs* között a legnagyobb különbség a filozófiájukban rejlik: a *xiafs* a minix-1 alapján épült, azon minél kevesebbet változtatva, hogy egy stabil rendszer legyen. Az *ext2fs* sokkal nagyratörőbb terv volt, amelynek elsődleges célja az volt, hogy sok lehetőséget adjon, és hagyjon helyet továbbiak számára is. Az *ext2fs*-hez például fejlesztés alatt áll egy undelete algoritmus – ez a Unix-világban teljesen egyedülálló lenne. Bár több mérés azt mutatja, hogy az *xiafs* gyorsabb, viszont az *ext2fs* sokkal kevésbé érzékeny a töredezésre. Az *xiafs* már stabil, megállapodott rendszer, az *ext2fs* – mint az egész Linux rendszer – folyamatos bővülésen megy keresztül. E bővülés igazi érdekessége, hogy a régi file-ok törlése nélkül illeszthető a rendszerbe a módosított file-rendszer.

Könnyen lehet, hogy mindkét file-rendszer – a jelenlegivel való kompatibilitást is beleértve – mintegy 2 Tbyte-os (2048 Gbyte) méretű partíciókat is lehetővé tesz (ez feltehetően néhány évig azért elég lesz). Az *xiafs* maximális file-név hosszt bármikor 255-re lehet emelni, ehhez mindössze egyetlen konstanst kell megváltoztatni, és újrafordítani a kernelt. Az *ext2fs* tervezett jövőjében szerepel még a tömörített file-rendszer, az undelete, és a superblock sérülése esetén történő helyreállítás is.

## DOS-alapú file-rendszerek

A Linux természetesen támogatja a DOS-t is, kétféleképpen: az egyik a DOS emulátor (DOSEMU) és a DOS file-rendszer támogatása. Ezt *dosfs*-nek hívják.

Használatával írhatók-olvashatók a FAT-os partíciók és floppyk file-jai. A Linux természetesen támogatja a FAT kiterjesztéseit is, mint a VFAT, illetve a FAT32 támogatás is fejlesztés alatt áll.

A DOS FAT rendszerére épül egy érdekes file-rendszer, az *UMSDOS*. Ez egy DOS-partícióban tud Linux rendszernek megfelelő file-rendszert létrehozni, a már ott levő DOS rendszer megbolygatása nélkül. Képes tárolni 255 karakteres file-neveket és a három időmezőt. Ezt a több DOS-kiegészítés – elsősorban a 4DOS – által használt módszer teszi lehetővé: egy rejtett file-ban tartja a kiegészítő információkat. E file-rendszer előnye, hogy a Linux rendszert újraparticionálás nélkül is használhatjuk. (E fejlesztéshez kiválóan kapcsolódik a Linux Loader – *loadlin* –, ami egy DOS-parancsra „berántja” a Linux OS-t, valamint a DOSLinux, ami egy lecsupaszított Linux (UMSDOS file-rendszeren.) Az *UMSDOS* legfőbb hibája a lassúsága, hiszen mint az elején írtuk, a FAT rendszer nem rakéta...

## procf, a rejtőzködő

Van egy nagyon különös, leginkább technikai értelemben fontos file-rendszer. Ezzel az „egyszeri” felhasználó nem nagyon fut össze. Bár folyamatosan használja, mégsem kell tudnia, hogy ott van. Ez egy különleges, a Unix-tradíciókkal szakító file-rendszer. A szokványos módszer a processzek adatainak tárolására: egy file a /dev könyvtárban. Ehhez közvetlenül kell hozzáférni, hogy infót szerezzünk a processzekről, ami veszélyes és felesleges.

A Linux *procf*-t használ: egy külön file-rendszerben tárolja az ilyen infókat. Ez sokkal-sokkal tisztább megoldás, egyszerűbb is nyomom követni, mi történik. A felhasználók ebből nem sokat látnak, de a kernel annál többet.

## Készítsünk saját rendszert!

Végül a bővítésről: Hogyan adhatunk hozzá file-rendszereket a Linuxhoz? Ez nem könnyű, mivel a tradíció szerint ezek a kernel részei. Azonban megjelent már a *userfs*, és ez a haladás iránya is: a felhasználók által betehető file-rendszerek. A Linux alatt már most is vannak ilyenek (pl.: a csak olvasható OS/2 HPFS és Windows NT NTFS), de az igazi a GNU-project részeként készülő Hurd lesz – ebben minden, de minden cserélhető lesz a felhasználó által... Egy ilyen file-rendszer jó példa még a CFS (Cryptographic Filesystem), ami egy titkosított file-rendszer. ■

Szabó Balázs  
(dlux@tarki.hu)

# Az X-Window System

**A Linux-felhasználóknak sem kell nélkülözni a könnyen használható, grafikus kezelői felületet. Ez az X-Window. E cikk témája az, hogyan konfigurálhatjuk a Linuxunkon meglévő X-rendszert, illetve hogyan alakíthatjuk olyanná, amilyenné szeretnénk.**

Az X-Window System első széles körben használt verzióját, az X10R4-et 1986-ban a MIT-en (Massachusetts Institute of Technology) fejlesztették ki. Az elképzelés az volt, hogy egy hálózat számára átlátszó, hálózatiorientált, hierarchikus, szabványos, nyitott, eszközfüggetlen, színes grafikát támogató rendszert tervezzenek. Az X10R4 után már szinte évenként jöttek ki az új változatok, az első az X11R1 volt, amely nem volt kompatibilis az X10-ekkel, minden alkalmazást újra kellett fordítani, hogy X11 alatt menjen, mindennek ellenére még népszerűbb lett, mint elődjei. Az X11R2, R3, majd a későbbi verziók elsősorban javításokat tartalmaztak az előzőekhez képest, de újítások is voltak bennük. Pl.: az X11R5-ben vezették be a fontserver fogalmát, az X11R6-ban a Kerberost használó azonosítást. Ma az X11R6.3-nál tartanak.

## A Linux X-e

A Linuxban megtalálható X-támogatást az Xfree86 Projectnek köszönhetjük, akik megpróbálták az X Consortium által megalkotott szabványt x86-os Linux rendszerekre átültetni. Az XFree86-ból a mai legújabb verzió a 3.3, amely már számtalan videokártyát támogat, nagyon sokat közülük gyorsítófunkciókkal is.

Nem referencia jellegű leírást szeretnék, inkább csak ízelítőt, de mindenütt szeretném megjegyezni, hogy hol lehet a pontos szintaxishoz és teljes leíráshoz jutni. A példák megértéséhez szükséges a bash shell programozásának alapszintű ismerete. Mivel nagyon sokféle

disztribúciója van a Linuxnak, ezért a konfigurációs file-ok helyei eltérhetnek egymástól. Én elsősorban a RedHat 4.1-ben szereplő példákkal fogok élni, ami pl. Debian alatt szintén ugyanott van, viszont mások az alapértelmezései.

Az X-rendszer képernyőt kezelő része, amellyel a felhasználó közvetlenül kommunikál, az a szerver – a képernyőn levő ablakok, a képernyőhöz kapcsolódó programok pedig a kliensek. Mivel a rendszer a hálózati számúra átlátszó, ezért nem feltétlenül kell egy programnak azon a gépen futnia, amelyiken az ablaka megjelenik, a kettő közti kommunikáció hálózaton keresztül is megtörténhet, az ún. X-protokoll felhasználásával.

A szabvány nem tartalmaz ablakkezelő startegiót, tehát nem írja elő, hogy milyen ikonok legyenek az ablakkeretekben, milyen egérgombnyomásra mi történjék, stb. Ezek ellátása egy speciális kliens, a window-manager (ablakkezelő) program feladata. Egy átlagos kliens (nem window-manager) kap egy ablakot a képernyőn, azon új ablakokat hozhat létre, önmagán belül bármit megjeleníthet, de az önmagán kívüli területet nem változtathatja meg. A window-manager ezen kívül még megteheti azt, hogy az ablakot egy meghatározott kerettel veszi körbe, és a képernyő egész részét birtokol(hat)ja. Logikus, hogy ebből maximálisan csak egy lehet a képernyőn. Annak a felületnek az egészét, amellyel a rendszer és a felhasználó érintkezik, displaynek nevezzük. Könnyű összekeverni ezt a screen fogalmával, ami a megjelenítőeszköz neve (monitor). Egy displayhez több screen is tartozhat (*vagyis gyakorlatilag a munkaterületünk egyszerre több monitoron is elhelyezkedhet – a szerk.*), ezt a rendszer megengedi, de otthoni használatban ez általában nem szokott előfordulni. A display része még egy billentyűzet és egy „pointing device”, vagyis mutatóeszköz, ez általában az egér vagy a trackball.

A Linuxban az X-es dolgokat a /usr/X11R6 könyvtárban találjuk meg, de ezeket szokták /usr/X11, /usr/X386 könyvtárakra is linkelni. A RedHat 4.1-ben viszont a /usr/X11 nevezetű link hiányzik, ezért ezt érdemes kézzel létrehozni, hátha lesz olyan programunk, ami ezt keresi (ln -s /usr/X11R6 /usr/X11). Ezen belül elsősorban a bin könyvtárra érdemes felhívni a figyel-

met, ahol a futtatható állományok vannak, majd a lib/X11 alkönyvtárra, ahol a különböző konfigurációs file-oknak van a szabványos helye. Ez utóbbi könyvtár (teljes elérési úttal: /usr/X11R6/lib/X11) megtalálható a /etc/X11 könyvtárban is, azért linkelték ide, hogy könnyebben elérhetőek legyenek az adatok. A cikk hátralevő részében főleg ezt a könyvtárnevet fogjuk használni. Régi rendszereken (pl. Slackware 3.0) ez a link nem volt meg, illetve nagygépes operációs rendszereken (Solaris, OSF) szintén csak a /usr/X11/lib/X11 (Solarisnál /usr/openwin/lib/X11) található meg. A Linuxhoz jelenleg 13 különböző szerverprogram létezik, ezek közül a legtöbb videokártyát támogató program az XF86\_SVGA, a többi általában egy gyártó egy-egy típusú kártyáinak a vezérlését oldja meg (pl.: S3, Mach8, Mach32, S3Virge, 8514 stb.). A 13 féle szerver közül telepítéskor általában eldől, hogy melyik kell, így a rendszer azt beállítja, azontúl X néven lehet rá hivatkozni (ez lesz az alapértelmezett X-szerver). Annak beállítására, hogy egy szerver felismerje és kezelje a megfelelő hardvert, az *xf86config* vagy a legújabb verziókban az *XF86Setup* nevezett program való. Ez rákérdez a monitor típusára, horizontális, vertikális időzítési információira, így ha maximálisan szeretnénk kihasználni a monitorunk és videokártyánk adta lehetőségeket, akkor érdemes utánanézni a megfelelő értékeknek. Ezeknek a programoknak az eredményeképpen létrejön a /etc, vagy a /etc/X11 könyvtárban egy *XF86Config* nevezett file, mely az általunk beállított értékeket fogja tartalmazni. Ha nem Xfree86-ot szeretnénk használni X-szervernek, hanem más, kereskedelmi forgalomban kapható X-szerverünk van (AcceleratedX, MetroX stb.), akkor annak a telepítése és konfigurálása természetesen más módon történik, erről a szoftverhez adott dokumentáció általában kielégítő információt nyújt. A sikeres beállítás után az X-szerver az „X” paranccsal érhető el és paraméterezhető.

Tehát megvan az X-szerverünk, de még nem tudjuk használni. Kétféle indítási módja van, az egyik az *xinit*-en keresztül, a másik pedig az *xdm*-es.

## Az xinit

Az xinit-es indítást akkor érdemes használni, ha a gép nincs hálózatba kötve, és a karakteres terminált részesítjük előnyben, csak néha folyamodunk grafikus felülethez. Ilyen esetben az indítás egy *startx* nevezett script segítségével történik, amely a /usr/X11R6/bin-ben van. Ezt jobban szemügyre véve könnyen átlátható, hogy hogyan működik. Az első néhány sorban beállítja, hogy mik azok az alapértelmezett file-ok, amiket

használ, majd megvizsgálja ezek létezését, utána a paramétereknek megfelelően összeállítja az xinit program paramétereit, és meghívja az xinit-et. Az xinit-nek 2 paramétere van. Az első azt a paraméterlistát tartalmazza, amit a szervernek ad át, majd egy „-” következik, és utána a kliensnek átadott paraméterek. Mind a kettő egy-egy programnak a neve és ennek paramétere. Ezek – ha léteznek a felhasználó HOME-jában –, akkor a .xinitrc és a .xserverrc, ha ezek nem léteznek, akkor a /etc/X11/xinit könyvtárban levő xinitrc ill. xserverrc, ha ezek sem léteznek, akkor egy olyan alapértelmezéssel indul el a rendszer, melyben a szerver a paraméterek nélkül megadott „X” parancs lesz, a kliens pedig egy xterm program. Általános esetben a /etc/X11/xinit/xinitrc indul el, a szerver pedig paraméter nélkül fut. Nézzük most meg a /etc/X11/xinit/xinitrc-t!

Először a közös és a felhasználó által használt erőforrás- és billentyűzet-file-ok neveit állítja be, majd ezeket sorrendben betölti, ha léteznek. Az erőforrás-file-ok a .Xresources, illetve a /etc/X11/xinit/.Xresources, ezek betöltését az *xrdb* program végzi, a billentyűzet-kiosztást (.Xmodmap, /etc/X11/xinit/.Xmodmap) az *xmodmap* parancs tölti be. Ezután megvizsgálja, hogy létezik-e egy .Xclients nevű file, amely tartalmazza a még elindított programokat, ha nem, akkor indít egy *xclockot*, 2 *xtermet*, egy *arenát*, amellyel a RedHat gégről és a Linuxról kaphatunk információkat, majd végül elindítja az *fvwm*, illetve ha ezt nem találja, akkor a *twm* ablakkezelőt. (A különböző ablakkezelőkről további cikkeinkben még szó lesz – a szerkez.)

Alapértelmezésben minden felhasználó megkapja a .Xclients file-t egy alapbeállítással, ami abból áll, hogy a TheNextLevel, az fvwm95, az fvwm2 és az fvwm ablakkezelők közül elindítja azt, amelyik elérhető, ezen kívül nem indít alkalmazást. Néhány példával szeretném illusztrálni a fent leírt konfigurációkat:

## Példák az xserverrc és az xinitrc beállítására

Ha azt szeretné egy felhasználónk, hogy az X-szerver mindig 16-bites színmélységben induljon el az ő számára, akkor a .xserverrc-be azt kell beleírni, hogy:

```
#!/bin/bash
```

```
exec X -bpp 16 $*
```

Azért *exec*, hogy a memóriában ne maradjon benn a *bash*, ami a .xserverrc-t futtatja, és azért kell a \$\* a végére, hogy továbbadja a szervernek az egyéb paramétereket is, amelyeket például a *startx*-nek megadha-

tunk. Ha ugyanazt egy mindenki számára érvényes alapértelmezésként szeretnénk beállítani, akkor ezt a bejegyzést a `/etc/X11/xinit/xserverrc`-be kell tenni, amit minden felhasználó felülbírálhat egy saját `.xserverrc` segítségével. Ha azt szeretnénk, hogy egy felhasználónak az eddigi programok mellett még induljon el egy `xearth` is, akkor azt a `.Xclients`-be kell beleírni, lehetőleg az első sorok valamelyikébe, `&`-jellel a végén (mivel utána még indítani fogunk programot, és nem várunk addig, míg ez befejezi a futását). Ha saját billentyűzet-definíciót szeretnénk, akkor azt a `.Xmodmap` file-ba tessük, ekkor a rendszer induláskor ezt fogja nekünk betölteni. Saját billentyűzet készítéséhez az `xkeycaps` nevezetű program ajánlott, ami megszerezhető pl. az `ftp://sunsite.unc.edu/pub/Linux/distributions/redhat/contrib/contrib-2.0.x/RPMS` könyvtárból, vagy hazai mirrorokról is. Bátrabbak nekiállhatnak `xmodmap`-p-k vagy a man `xmodmap` alapján dolgozni.

## Az X Display Manager

Az `xdm` (X Display Manager) egy alternatívája a karakteres `init/getty/login` programoknak. Amikor elindítjuk, akkor egy grafikus képernyőt kapunk, a közepén egy `Login`: és egy `Password`: mezővel. Miután sikeresen bejelentkeztünk, hasonló kép tárul elénk, mint azt a `startx`-nél leírtuk, majd kilépéskor újra visszakapjuk a `login`-ablakot. Emellett a lehetőség mellett az `xdm` elsődleges szolgáltatása az, hogy ezt az ablakot nemcsak a lokális számítógépre, szolgáltatóra, hanem bármely, ugyanazon a hálózaton elérhető számítógépre is, ezek a gépek ekkor X-terminálok. Ha egy `A` jelű gépen elindítottak egy `xdm`-et, akkor azt egy `B` jelű gépről el tudjuk érni ugyanúgy, ahogy például a telnet megy a két gép közt. A `B` gépen ehhez egy „`X -query A`” parancsot kell végrehajtani, ekkor ugyanazt a grafikus `login`-ablakot kapjuk meg, mint ami az `A` gépnél található `xdm`-indításkor.

Az `xdm` a `queryn` kívül nyújt még egy ún. `indirect` szolgáltatást is, ez az „`X -indirect`” parancsral próbálható ki. Ekkor beállíthatóan egy menüt ad (choosert), amelyben választhatunk hostok közül, hogy melyiktől szeretnénk igénybe venni a `query`-szolgáltatást, vagy kérdés és menü nélkül meghívhat egy `queryt` egy másik szerver `xdm`-jére. A menü természetesen konfigurálható, mint ahogy az `xdm` nagyon sok egyéb része is.

Az `xdm` indítása több módon is történhet. Ha a `/etc/inittab`-ból indítjuk, (ott van már, csak ki van kommentezve), akkor azt az `init` mindig újraindítja, ha véletlenül valamilyen programhiba folytán, vagy a felhasználó akaratából „meghalna”. Hátránya, hogy ha szeret-

nénk sem tudjuk letiltani a gép üzemeltetése közben, csak a gép újraindításával. Ha a `/etc/rc.d/rc.local`-ból, vagy más `rc`-file-okból indítjuk, akkor az eredmény ugyanaz, de később bármikor kilöhetjük (pl.: „`killall xdm`”). Az `xdm`-et inittabból a `-nodaemon` opcióval kell indítani, mivel az `init` az indított file-okat mindig újraindítja, amint kiléptek (`espawn` opció az inittabbban), és ha nem adjuk meg, akkor az `xdm` forkol és kilép, az `init` pedig azt hiszi, hogy ekkor befejeződött a futása, tehát ismét elindítaná. A `-nodaemon` kapcsoló használata mellett nem teszi, csak akkor lép ki, ha kapott valamilyen szignált. Tehát ha inittabból szeretnénk használni az `xdm`-et, akkor az utolsó sorba az

```
„xd:35:respawn:/usr/bin/X11/xdm -nodaemon”
```

bejegyzést kell tenni. Ekkor a 3-as és az 5-ös runlevelen betölti az `xdm`-et. Megadhatunk neki konfigurációs file-t is (`-config file-név`), ez alapértelmezésben a `/usr/X11R6/lib/X11/xdm/xdm-config (/etc/X11/xdm/xdm-config)`.

Az `xdm-config` erőforrás adatai X-resource formátumban vannak: egy sorban a kettőspontig van a resource neve, utána az értéke. A resource neve pontokkal tagolható részekre. A resource nevében gyakran használatos a „`*`” karakter helyettesítésre. Az `xdm`-re vonatkozó resource-ok a „`DisplayManager.opció`”, az `xdm` egyes displayeire vonatkozó adatok „`DisplayManager.display.opció`” részben találhatóak meg. A display itt annak a displaynek a neve, ahonnan a query vagy az `indirect` kérelem jött (ahol az X-server van). Mivel a display-név tartalmaz(hat) pontokat (pl.: `one.two.hu:0`), ezért itt a pontot és a kettőspontot az aláhúzás helyettesíti (`one_two_hu_0`). Néhány beállítható resource az `xdm-config`-ban:

- **DisplayManager.servers**: az `xdm` által indított szervereket leíró file nevét adhatjuk meg, vagy ha egy „`/`”-jellel kezdődik a sor, akkor egy szerverleírást tartalmaz (bővebben később). Az alapértelmezés a `/etc/X11/xdm/Xservers` file.

- **DisplayManager.errorLogFile**: a file neve, ahova az `stderr`-átírányítja az `xdm` a scriptjeitől (`Xstartup`, `Xsession`, `Xreset`, `Xsetup`). Alapértelmezés: `/etc/X11/xdm/xdm-errors`.

- **DisplayManager.debugLevel**: ugyanaz, mint az `xdm` -debug opciója. Ha nem 0 az értéke, akkor a daemon módot letiltja, és bővebb információt szolgáltat az `xdm` futásáról. Hibakereséskor fontos opció. Üzenetei értelmezéséhez szükség lehet az `xdm` forráskódjára is.

- **DisplayManager.removeDomainname**: amikor a display nevére van szüksége az `xdm`-nek (pl.: a resource-ok meghatározásánál), false-ra állítja a teljes nevet (fully qualified domain name) használja a gépnek, true-ra állítva a rövidebbet (vagyis a domain név nélkül).

● *DisplayManager.accessFile*: a hozzáférések és szolgáltatások leírását tartalmazó file (/etc/X11/xdm/Xaccess) elérési útja. Ebben a file-ban kell megadni, hogy mely hostok jogosultak query és indirect hozzáférésre. Alapértelmezésben a query mindenhonét elfogadott, indirectet viszont nem fogad el. Az Xaccess file-ban van leírva, hogyan állíthatók a hozzáférések, néhány sor kommentezésével ez könnyen el is végezhető.

A következő erőforrások egy-egy displayhez kapcsolódhatnak, általában a display neve helyett „\*” van, de bármely displaynek készíthetünk ezekből egyedít is.

● *DisplayManager.display.chooser*: az indirect kérelemnél meghatározza, hogy mely program legyen a chooser, alapértelmezése: /etc/X11/xdm/chooser.

● *DisplayManager.display.setup*: annak a programnak (általában shell scriptnek) a neve, ami rootként fut le a login-ablak megjelenése előtt. Alapértelmezés: /etc/X11/xdm/Xsetup.

● *DisplayManager.display.startup*: a sikeres bejelentkezést követő script, ez is rootként fut le. Ha nincs semmi megadva az xdm-config-ban, akkor nem fut le semmi, a RedHat Linux alatt a /etc/X11/xdm/TakeConsole script fut le lokális bejelentkezés esetén (:0 display).

● *DisplayManager.display.session*: annak a scriptnek a neve, ami a felhasználónak betölti az X használatához szükséges programokat, és addig fut, amíg a felhasználó ki nem lépett. Ez a script a startup script után indul, de már a felhasználó jogaival. Ha nincs megadva itt semmi, akkor egy /usr/X11R6/bin/xterm indul el, de általában például a RedHatnál a /etc/X11/xdm/Xsession script szokott itt állni.

● *DisplayManager.display.reset*: a session után lefutó script, amely visszaállít esetlegesen a képernyőn elállított dolgokat. Ez is rootként fut, a RedHatnál ez alap esetben a /etc/X11/xdm/GiveConsole, ami a /dev/console jogait állítja vissza a rootéra. Nem lokális X esetén a RedHat nem használja. Lefutása után a login-ablak jön vissza (Xsetup).

● *DisplayManager.display.systemShell*: meghatározza, hogy a display reset és startup scripteit milyen shell futtassa az xdm. Az alapértelmezés: /bin/sh.

● *DisplayManager.display.resources*: a login-ablak számára szükséges X-resource-okat tartalmazó file neve. Ebben tudjuk meghatározni pl.: a login-ablak méretét, pozícióját, betűkészletét stb. Az alapértelmezés a /etc/X11/xdm/Xresources.

Az xdm indításakor elindítandó lokális X-szervereket konfigurálhatjuk a /etc/X11/xdm/Xservers file-ban (illetve az xdm-configban átírhatjuk másra). Alapértelmezésben a :0 képernyőn indít egyet, ha egy „#” jelet te-

szünk a :0 kezdetű sor elé, akkor ezt nem teszi, de a többi funkciója (query, indirect kiszolgálás) megy. Ha több szervert is szeretnénk felvenni, akkor a :0 mellé fel lehet venni például egy :1-t, ami más paramétereket kap (pl.: más virtuális felbontást, vagy 16 bites színmélységet használ stb). A paraméterek formátuma: „:<display-szám> local <indítandó program neve és paraméterei>”. A fent említett 16 bites színmélység és egy alternatív konfigurációs file használata például így néz ki: „:1 local /usr/X11R6/bin/X -xf86config/etc/X11/XF86config.virtual -bpp 16”.

A DisplayManager.display.resources által mutatott file-t használja a rendszer a login-ablak erőforrásainak beállítására. A login-ablakra az *xlogin*mal kezdődő resource-ok vonatkoznak. Beállítható, hogy milyen fonttal legyen kiírva a szöveg (*xlogin\*Font*), mi legyen a bejelentkező felirat a login-ablak tetején (*xlogin\*greeting*) stb. Az összes szöveget át lehet írni ezen keresztül, a login-ablak pozícióját (*xlogin.Login.x*, illetve *xlogin.Login.y*), az ablak méretét (*xlogin.Login.width*, *xlogin.Login.height*). A login-ablakon kívül ide kerülnek az *xconsole* és a *chooser* resource-ai is.

A /etc/X11/xdm/Xsetup\_0 file RedHat-ben csak a konzolnál fut le, más helyről használva a queryt, nincs definiálva funkció a login-ablak megjelenése előtt. Amennyiben valamilyen képet szeretnénk betölteni az ablak előtt, úgy érdemes csinálnunk egy /etc/X11/xdm/Xsetup file-t, és az xdm-configba beírni egy „DisplayManager\*setup: /usr/X11/lib/X11/xdm/Xsetup” sort a végére. Ekkor minden nem lokális szervernél lefut. Az Xsetup file-ba pedig beírni például:  
#!/bin/bash

```
xloadimage -onroot -fullscreen -smooth
bejelentkezo_kep.jpg
```

Ekkor a login-ablak kirakása előtt megjelenik a kép. Helyezzük át a login-ablakot lehetőleg úgy, hogy a kép érdekes részeit ne takarja el a login-ablak (Xresources file).

A belépéskor használható billentyűkombinációkat szabályozhatjuk az Xresources file Xlogin.Login.translations: részében. Általában erre nincs szükség, aki szükségét érzi annak, hogy az alapértelmezéseket átírja, az a „man xdm”-ben megtalálja, hogy milyen lehetőségei vannak.

A startup script feladata, hogy megvizsgálja azokat a dolgokat, amelyeket a login programnak kellene megvizsgálnia tlnet/rlogin esetén (/etc/nologin létezése esetén a belépés letiltása stb.), illetve hogy az Xsetup által felrakott kép – ha van – erőforrásait (színek) felszabadítsa (pl.: Xbanner csomag, freetemp program). A RedHatban nem használják ki ezt az opciót, tehát xdm-en

keresztül akkor is be lehet lépni, ha a telnet már nem enged be. Ez nem feltétlenül előny, ezért gyorsan megírható egy olyan Xstartup program, ami ezt az ellenőrzést elvégzi:

```
#!/bin/bash

if [ -f /etc/nologin ]
then
    xmessage -file /etc/nologin -timeout 30 -center
    exit 1
fi
```

A session program már a felhasználó jogaival fut, ha a script a végére ér, akkor azt a rendszer úgy fogja fel, hogy a felhasználó kilépett. Az alapértelmezett script normális esetben megnézi, hogy van-e a felhasználó home-jában .xsession, és azt indítja el, ha nem létezik, akkor a .Xresources file-ból betölti a felhasználó által definiált erőforrásokat, illetve elindítja az x session managert (xsm). A .xsession létezése esetén ezek a programok futnak le. Mivel a .Xclients-re link a .xsession, ezért a session addig tart, amíg a korábban leírt .XClients scripts fut. Ez indítja a window-managert, meg a járulékos programokat.

Megjegyzendő, hogy az alapértelmezés hibája az a feltételezés, hogy a display az :0.0, tehát ahhoz, hogy más displayról használjuk, mindenképpen át kell írni a .xsession-t. Ugyanilyen hiányosság, hogy míg xinit-es indításkor betölti a .Xmodmap és a .Xresource file-okat minden esetben, itt ezt nem teszi meg (nekünk kell a .xsession-ban).

Végezetül álljon itt néhány programtöredék, amely még jól jöhet a scriptekbe.

- Az alapértelmezett .Xsession scriptből kihagyták az alapértelmezett billentyűzet- és erőforrás-file-ok betöltését:

```
if [ -f .Xresources ]
then
    xrdm -merge .Xresources
fi

if [ -f .Xmodmap ]
then
    xmodmap .Xmodmap
fi
```

- Ha azt akarjuk, hogy a felhasználó saját .xsession file-ja egy kis ablakban fusson, akkor a /etc/X11/xdm/Xsession file-ba beírandó az exec .xsession helyett:

```
$usersession=$HOME/.
Xsession;
```

```
$systemsession=/etc/X11/xdm/Xsession.default
if [ -f $usersession ]
```

```
then
    session=$usersession;
else
    session=$systemsession
fi

exec rxvt -geometry 80x5+0-36 -T 'LOGIN-CONSOLE' \
    -C -e bash -login "$session"
```

- Az Xsession.default file pedig legyen egyszerűen csak annyit, hogy elindít egy fvwm window-managert, és egy shellt az ablakban:

```
#!/bin/bash
echo 'Ebből az ablakból lép ki utoljára'
fvwm&
exec `cat /etc/passwd | grep \^whoami\` | cut -f 7 -d ':'`

● Ha a rendszerben van NIS (yellow pages), akkor az utolsó sor egy picivel bonyolultabb:
exec `ypcat passwd | cat - /etc/passwd | grep \
    "\^whoami\"` | tail -n1 | cut -f 7 -d ':'`
```

- Mivel a window-managert itt már egy login-shell indítja, nem az X, ezért a window-manager lefutásakor már lefutott a felhasználó profile-ja, a rendszer profile-ja, a bejelentkező üzeneteket és az egyéb hibákat a LOGIN-CONSOLE-on láthatjuk, és ha szeretnénk window-managert váltani működés közben, akkor nyugodtan kilöhetjük a window-managert anélkül, hogy véget érne a sessionünk, majd a login-console-ból indíthatunk egy másikat.

Persze a felhasználónak már létezhet saját .xsession-ja, de javasolhatunk neki is hasonló megoldást. Egy felhasználó számára elkészített .xsession file kb. így nézhet ki:

```
#!/bin/bash
# indítsunk 2 x-termet
xterm &
xterm &
#általában CD-t is szoktunk hallgatni
xplaycd &
#majd indítsunk egy window-managert
fvwm &
#a végén legyen egy shell-ünk a LOGIN-ablakban.
exec bash
```

- Érdemes beletenni a /etc/X11/xdm/Xsetup-ba, hogy a BackSpace és a Delete ne ugyanazt a funkciót jelölje, mivel alapértelmezésben problémák szoktak adódni ebből:

```
/usr/X11R6/bin/xmodmap -e 'keycode 22 = BackSpace'
```

\* \* \*

Röviden ennyi fért bele az X software környezet konfigurálásával foglalkozó részbe. Ez alapján igényes X-es felületet adhatunk gépünknek. A window-managerek konfigurálása külön cikkekből olvasható. ■

Kégl Tamás  
(monty@python  
.mars.vein.hu)

# Az Fvwm ablakkezelő

Az alábbi és az elkövetkező három cikkben az ablakkezelők világába vezetjük be az Olvasót. A leírás célja, hogy lehetőleg minél átfogóbb betekintést nyújtson X-es környezetünk hatékonyabb és szebb beállításához. Ezáltal nem csak a „fapados Unix” tévhitet feledhetjük el, de munkánk hatékonysága, sebessége is lényegesen jobb egy testre szabott, logikusan beállított felület mellett.

Az ablakkezelő (window manager) egy olyan program, ami az ablakok elhelyezkedését, tulajdonságait szabályozza, lehetőséget biztosít a felhasználónak, hogy az ablakokat mozgassa, átmeretezze, behajtsa (ikonba), a többi ablak mögé vagy elé tegye. Egy ablak a tartalmán kívül általában a következő részekből áll (melyeket dekorációnak is szoktak hívni): fejléc valamilyen címkével, nyomógombokkal (melyekhez különféle funkciók rendelhetők), keret, a keret sarkai, melyek általában külön kezelendők magától a kerettől. Különlegesen számít a háttérablak, amelynek nincsenek dekorációi, viszont az egérgombok lenyomásához különféle menük rendelhetők. A modernebb ablakkezelők többnyire tartalmazhatnak egy vagy több speciális ablakot is, melyek ikonjaik segítségével programok kattintással történő elindítását teszik lehetővé, esetleg menük hívhatók elő segítségükkel.

Az idők során számtalan window manager jött létre, köztük megemlítendő a twm (Tab Window Manager) – ez az egyik legrégebbi, mely az X-Window rendszer része –, az OSF/Motif részét képező mwm, a Sun és az AT&T által kifejlesztett olwm (Open Look Window Manager), és a linuxosok által leggyakrabban használt Fvwm. E lap-

ban csak az Fvwm-mel és két legelterjedtebb származékával (fvwm95 és AfterStep) fogunk foglalkozni, majd a jövő egyik lehetséges változatát mutatjuk be.

## Az Fvwm 1.24

Az Fvwm ablakkezelő a twm bázisán alakult ki, annak alapos áttevésével. Ennek köszönhetően a memóriaszükséglete lényegesen kisebb lett, ugyanakkor a háromdimenziós ablakok kelleme- sebb megjelenést nyújtanak.

Először egy kis szómagyarázat következze, hogy mit jelent az Fvwm. Nos, ez egy rövidítés, a teljes nevén F(?) Virtual Window Manager. Bővebb magyarázatot az első két tag érdemel. A szerző, Robert Nation, állítása szerint már elfelejtette, mit is jelent az F. Néhány lehetséges magyarázatot a FAQ-ból: Famous (híres), Fantastic (fantasztikus), Flexible (hajlékony), Fast (gyors), Fvwm (GNU-féle rekurzív definíció).

A második tag a virtuális jellegre utal. A virtuális itt azt jelenti, hogy a teljes munkaterületünk több látszólagos képernyőből épül fel, és ezek között többféle módszerrel tudunk navigálni.

## Főbb beállítási lehetőségek

Az Fvwm a többi ablakkezelőhöz hasonlóan egy olvasható – és módosítható – szövegfájlban tartja beállításait. CD-nken több ilyen file-t is adunk mintául, természetesen mindenki kedve szerint átkonfigurálhatja ezeket. Az 1.24-es Fvwm lehetőségeit is egy ilyen file alapján mutatjuk most be:

**Fontkezelés:** A *Font* utasítással lehet a legördülő menükhöz tartozó fontokat beállítani. A *WindowFont* az ablakok, az *IconFont* az ikonok feliratainak kinézetét szabja meg.

```
Font          -adobe-helvetica-medium-o-*-*-16-*-*-*-*-*
WindowFont   -adobe-helvetica-bold-o-*-*-15-*-*-*-*-*
IconFont     fixed
```



## GOMBOK KÓDOLÁSA

Cursor 3 60

ButtonStyle : 1 5 20x40@1 80x40@1 80x60@0 20x60@0 20x40@1

ButtonStyle : 2 5 20x20@1 20x80@1 80x80@0 80x20@0 20x20@1

ButtonStyle : 4 5 40x40@1 60x40@1 60x60@0 40x60@0 40x40@1

ButtonStyle : 6 13 26x29@1 34x21@1 50x35@1 70x21@1 79x29@1 63x48@0 79x65@1 70x75@0 50x61@0 34x75@0 26x65@0 44x48@1 26x29@0

**Fokuszálás:** Itt nagyon eltérnek egymástól az egyéni ízlések. Én nagyon szeretem használni az *AutoRaise* funkciót, másokat viszont ez az örület-be tud kergetni. Arról van itt szó, hogy amelyik ablak felett az egérkurzor van, az ugrik legfelülre. Az utasítás utáni számmal lehet beállítani, hogy ez hány milliszekundum idő elteltével következék be. Például:

AutoRaise 750

Ha ez nem szimpatikus, még mindig lehetőség van arra, hogy az az ablak legyen aktív, amelyik felett az egérkurzor van (ez az alapbeállítás az SGI-féle 4Dwm-ben). Végül lehetséges az is, hogy az aktív ablakot (mint a Motif window managerben) rákattintással válasszuk ki:

ClickToFocus

További fókuszálási lehetőséget jelent a *SloppyFocus* opció, melynél a fókuszált ablak aktív marad, ha nem egy másik ablakba, hanem a root window fölé mozgatjuk az egeret. (A root window gyakorlatilag a háttérnek felel meg, vagyis annak, amin elhelyezkednek az ablakok, az ikonok, vagyis minden – a szerk.)

**Ikonok:** Az IconBoxok kijelölésével szabhatjuk meg, hogy desktopunk mely területére (netán területeire) kerüljenek az ikonok. Ha negatív számokat adunk meg, akkor azok a jobb, illetőleg az alsó oldaltól számítanak.

IconBox -110 40 -5 -140

IconBox 5 -140 -140 -5

A *SuppressIcons* parancs megadásával eltüntetjük ikonjainkat a képernyőről, ez a funkció elsősorban alacsony felbontás esetén lehet hasznos. Ilyenkor érdemes ügyelni arra, hogy valamelyik egérgombra vagy billentyükombinációra működésbe lépjen a *WindowList* (ez általában a jobb gombhoz szokott hozzárendelt lenni), melylyel ráugorhatunk futó taszkjaink közül arra, amelyikre akarunk. Ha a *StubbornIcons* opciót beállítjuk, akkor nem arra a desktopra ugrik alkalmazásunk, ahol éppen vagyunk, hanem mi

ugrunk arra a desktopra, amelyiken az alkalmazás elhelyezkedik.

A *Stickylcons* jópofa funkció. Ha virtuális desktopot váltunk, az ikonok jönnek velünk.

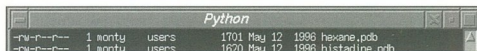
**MWM (Motif Window Manager) emuláció:** az *MWMFunctionHints*, *MWMDecorHints* és *MWMHintOverride* opciókat érdemes beállítani, elsősorban a motifos applikációk igényelhetik ezeket.

- *MWMMenus*: ha egy olyan menüre visszük rá a kurzort, amely alatt még almenü is rejlik, akkor azonnal előugrik, mint ahogy az a Motif-alkalmazásoknál szokásos.

- *MWMBorders*: a kereteknek élesebb az árnyékolása; 1 pixel széles a normál 2 pixel helyett. Ha ezt megadjuk, akkor az ablakjaink kb. így fognak kinézni:



Ellenkező esetben meg így:



**Egyéb apróságok:** a *ClickTime* funkcióval beállítható, hogy több egérr kattintás esetében mennyi ideig számítson az dupla, illetve tripla kattintásnak. (Általában a triplakattintással egy egész sort lehet kijelölni.) Az alapérték:

ClickTime 150

Az *OpaqueMove* opcióhoz tartozó szám (N) arra vonatkozik, ha egy ablak többet foglal el a képernyő területéből, mint annak N százaléka, akkor mozgathatás közben csak a keret változtatja a helyét, az ablak csak letétlkor kerül az új pozíció-

## MENÜK FELÉPÍTÉSE

```

PopUp "Ablakműveletek"
  Title "Ablakműveletek"
  Function "Mozgat" Move-or-Raise
  Function "Átméretez" Resize-or-Raise
  Iconify "Ikonizál"
  Stick "Képernyőhöz ragaszt"
  Function "Maximalizál" maximize_func
  Nop ""
  Destroy "Agyoncsap"
  Delete "Bezár"
  Nop ""
  Refresh "Képfrissítés"
EndPopUp
#(Itt az "Iconify", a "Stick" és a "maximize_func" parancsok
#ismételt #végrehajtásával
#az eredetivel ellentétes hatást érhetjük el.)
PopUp "Alkalmazások"
  Title "Alkalmazások"
  Exec "Ghostview" exec ghostview &
  Exec "XV" exec xv &
  Exec "Xman" exec xman &
  Exec "Xfm" exec xfm &
  Nop ""
  Exec "Netscape" exec /usr/local/netscape/netscape &
  Exec "Lynx" exec /usr/local/bin/lynx -bw -2 &
#Az utóbbi két esetben természetesen a valódi elérési utat kell
#beírni.
EndPopUp
PopUp "Játékok"
  Title "Játékok"
  Exec "Workman" exec workman &
  Exec "Xhextris" exec xhextris &
  Exec "Xtetrtris" exec xtetrtris &
  Exec "Xbill" exec xbill &
EndPopUp
PopUp "Kilépek"
  Title "Tényleg ki akarsz lépni?"
  Nop ""
  Quit "Igen!"
  Nop ""
  Restart "Inkább legyen az afterstep!" afterstep
  Nop ""
  Restart "Indítsd újra az Fvwm-et!" Fvwm
  Nop ""
  Nop "Meggondoltam magam!"
EndPopUp
PopUp "Programok"
  Title "Programok"
  Exec "Számológép" exec xcalc &
  Exec "Xterm" exec xterm -sb -sl 500 -j -ls -fn 7x14 -bg seashell2 &
  Exec "Óra" exec oclock -minute red -hour blue -bg black &
  Exec "Képernyővédő" exec xlock -nolock -nice 10 -install -mode random &
  Nop ""
  PopUp "Alkalmazások" Alkalmazások
  Nop ""
  PopUp "Játékok" Játékok
  Nop ""
  PopUp "Kilépek" Kilépek
  Nop ""
  Refresh "Képernyőfrissítés"
EndPopUp
Mouse 1 R A PopUp "Programok"
Mouse 2 R A PopUp "Ablakműveletek"
Mouse 3 R A WindowList

```

óba. Nagy értéket (50–100) csak gyors video-kártyához ajánlott használni! Egy célszerűen megválasztott érték:

OpaqueMove 20

Az *EdgeScroll* és az *EdgeResistance* a virtuális desktopok közti gyors váltást hivatott elősegíteni, azaz, ha a képernyő széléhez húzzuk az egeret, átugorhatunk a következőre, vagy odébb ugorhatunk. Példák:

EdgeScroll 100 100 (teljesen átvált a következő desktopra)

EdgeScroll 50 50 (Fél képernyőt arrébbugrik)

Az *EdgeResistance* első paramétere az időt jelenti milliszekundumban, amennyi elteltével megtörténik a képernyőváltás.

A második egy távolságot jelent pixelben, aminek a közelebről mozgatjuk az ablakot a képernyő széléhez, akkor a WM meggátolja annak átcsúszását a túloldalra, egészen addig, amíg a kurzor is át nem kerül. (Itt be kell valljam, hogy ettől az egész *EdgeScroll* funkciótól néha szálanként tudnám kitépni a hajam, így aztán ezt nem is használnom.)

Viszonylag elfogadható értékek szerintem:

EdgeScroll 50 50

EdgeResistance 500 200

Az a gond, hogy ha az első értéket alacsonyra állítjuk, akkor rengetegszer fogunk véletlenül ugrálni a desktopok között, ekkor jön az a bizonyos hajtépős effektus. Ha meg túl magasra, akkor felesleges az egész, jobban járunk, ha Ctrl+kurzornyíllal, vagy egérrel váltunk desktopot. Így az általam használt érték:

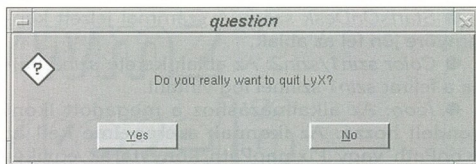
EdgeResistance 10000 0

**Ablakok elhelyezése:** ha nyitunk egy új ablakot, annak elhelyezése többféle módon lehetséges. Alapesetben a kinyíló ablakokat mi magunknak kell letenni egy kattintással, ha ezt el akarjuk kerülni, akkor a *RandomPlacement* opció használatát ajánlott.

Ha a *SmartPlacement* funkció be van kapcsolva, akkor az Fvwm megpróbálja fedetlen helyekre tenni az új ablakokat. Ha ez nem sikerül neki, akkor attól függően, hogy használunk-e *RandomPlacement*et, leteszi magától valahová, vagy ránk bízta a döntést.

Ha még *StubbornPlacement*et is használunk a *SmartPlacement* mellett, akkor az Fvwm az előpenderülő ablakokkal igyekszik nem letakarni az ikonokat.

**Decorációk:** *DecorateTransients* beállításakor az átmeneti jellegű ablakok (pl. dialógusdobozok) is kapnak dekorációt, például ilyet:



**Virtuális desktop beállítása:** szöveg helyett ismét egy példa. 3x3 egységnyi desktopot állítunk be, a területe a rootablak 28-ad részét foglalja el, és a (6,5)-es koordinátán van a bal felső sarka.

DeskTopSize 3x3

DeskTopScale 28

Pager 6 5

Az ablak fejlécében lévő nyomógomb kinézetét is természetesen szabadon variálhatjuk. Az előbbi példákban szereplő gombok kódjait az 1. képen tekinthetjük meg.

Az első sorban a kurzor típusát állítjuk be, hogy milyen legyen, ha épp gombjaink felett tartózkodunk. A 3-as szám a fejlécen lévő gombokra vonatkozik (van még ezen kívül 15 másik lehetőség, amelyek menüknél, a háttérablaknál, mozgatásnál, alkalmazások bezárásánál stb. érvényesek), a második a `/usr/include/X11/cursorfont.h` állományban definiált (nálam 78 féle) kurzor közül a 60-ast (kéz) rendeli hozzá a gombokhoz. A `ButtonStyle` kulcszót követően a gomb száma áll (bal oldalt a páratlan, jobb oldalt a páros számozásúak vannak kívülről befelé haladva), mögötte a mérete, majd a „vonalas rajz” következik. A `@1` jelzésű vonatkozik a világos, a `@0` a sötét színű vonalra.

**Stílusok:** a *Style* kulcsszóval az egyes alkalmazásokhoz rendelt erőforrásokat állíthatjuk be. Ha az alkalmazásnak „\*”-ot adunk meg, akkor az általános érvénnyel főg bírni. Ezen belül az alábbiak állnak rendelkezésre:

- **BorderWidth:** az átmeneti jellegű ablakok keretének vastagságát állítja be.

- **HandleWidth:** ugyanez az általános jellegű ablakokra. A keretvastagságok pixelben értendők.

- **NoTitle:** az ablakra nem tesz fejléct.

- **NoHandles:** az ablakot nem keretezi be.

- **Sticky:** odaragasztja az ablakot a képernyőhöz; ha virtuális desktopot váltunk, az ablak velünk jön.

- **WindowListSkip:** a megadott nevű ablakot kihagyja a felsorolásból, ha *WindowList*et kérünk.

- *StartsOnDesk* szám: A számmal jelzett képernyőre jön fel az ablak.

- *Color szín1/szín2*: Az ablak kerete *szín2*, rajta a felirat *szín1* színnel fog virítani.

- *Icon*: Az alkalmazáshoz a megadott ikonrendeli hozzá. Az ikonnak szerepelnie kell az *IconPath* vagy *PixmapPath* könyvtárak egyikében. Pl.:

```
Style "xterm" Icon xterm.next.xpm
```

## Függvények

**Function függvénynév**: összetettebb függvény definiálását teszi lehetővé az Fvwm beépített parancsai segítségével, melyeket aztán hozzárendelhetünk egérekattintás vagy billentyűzetkombinációkhoz.

A függvények a kiváltó események hatására hajtódnak végre, ezek a következők lehetnek: *Immediate* (azonnal végrehajtja a parancsot), *Motion* (ha a *ClickTime* opcióval megadott időn belül nem engedjük fel a lenyomott egérgombot, netán még mozgatjuk is), *Click* (kattintás) és *DoubleClick* (kettős kattintás). Itt aztán szabadjára engedhetjük fantáziánkat, a lehetőségek szinte korlátlanok!

A következő példában létrehozunk két függvényt, melyek az ablakok kinagyítását vezérlik.

```
Function "maximize_func"
    Maximize      "Motion" 0 100
    Maximize      "Click"  0 75
    Maximize      "DoubleClick" 92 75
```

```
EndFunction
```

```
Function "maximize_horiz"
```

```
    Maximize      "Click"  75 0
    Maximize      "DoubleClick" 96 0
```

```
EndFunction
```

Rendeljük hozzá ezeket a bal, illetve a középső egérgombhoz:

```
Mouse 1 2 N Function "maximize_func"
Mouse 2 2 N Function "maximize_horiz"
```

A következő fog történni: ha a bal gombbal a maximalizáló gombra kattintunk, az ablak függőlegesen megnő a képernyő magasságának 75%-ra. Ha rákattintunk és lenyomva tartjuk a gombot, az ablak a képernyő teljes magasságára kinagyítja magát. Kettős kattintásra vízszintesen is

megnő az ablak, majdnem a teljes képernyőt elfoglalja. (Hagy helyet esetleg a *GoodStuff*nak és az ikonoknak.) A középső gombbal vízszintesen tudjuk kinyújtani az ablakot háromnegyedrészt (szimpla kattintással), vagy majdnem teljesen (dupla kattintással).

Némi magyarázattal tartozom az egérgombok hozzárendelését illetően. A *Mouse* kulcsszót követően az első paraméter a gomb sorszámát jelzi. 1, 2, 3 jelentik a bal, középső, jobb gombokat, 0 esetében minden gomb ugyanazt a hatást váltja ki.

A második paraméter 0–9 számok esetében arra vonatkozik, hogy a fejlécen melyik gombot nyomtuk le. A gombok számozása megegyezik a *ButtonStyle* részben leírtakkal. Egyéb esetek:

**R**: root-ablak;  
**T**: az ablak fejléce;  
**W**: az alkalmazás ablaka;  
**S**: az ablak szélei;  
**F**: az ablak sarkai;  
**I**: ikonizált ablak.

Ezen környezeteket kombinálni is lehet, pl. az *FST* vonatkozik a teljes keretre fejlécestől, kivéve a fejlécen lévő gombokat.

A harmadik paraméter a valamilyen módosító-gombbal (gombokkal) együtt történő kattintásra vonatkozik. A módosítók lehetnek: *N* (semmilyen), *C* (Ctrl), *S* (Shift), *M* (Meta), *A* (ezek közül bármelyik).

## Menük felépítése

**Popup menünev**: definiál egy menüpontot, amit hozzárendelhetünk egy egérgombhoz, vagy egy billentyűzetkombinációhoz. A 4. képen olvasható példában definiálunk egy „Programok”, majd egy „Ablakműveletek” menüt, majd ezeket hozzárendeljük a bal, illetve a középső egérgombhoz.

A példa utolsó három sorában van a hozzárendelés. Pl.: a jobb egérgombhoz a *WindowList* funkciót rendeljük hozzá.

\* \* \*

Hát, ennyit az Fvwm 1.24 lehetőségeiről. A vége egy kicsit sűrűre sikeredett, de mivel még más ablakkezelőkről is szó lesz az elkövetkező cikkekben, az alapról talán elég ennyi is. ■

Kégl Tamás  
(monty@python  
.mars.vein.hu)

# Az Fvwm2 ablakkezelő

Az Fvwm 2-es verziója számos változtatást tartalmaz az 1.24-eshez képest. Sok parancs szintaxisa megváltozott, racionálisabb lett, sok új opció is felbukkant. Számos dolog menet közben is változtatható lett, a kinézet is igen tág teret enged fantáziánknak, ezáltal a konfigurációs állomány összetettsége szinte programozással felérő élvezetet nyújthat a felhasználó számára.

Természetesen a forráskód is meglehetősen szabadon „hackelhető”, ennek köszönhetően jöttek új window manager fejlesztések is, melyeket az Fvwm fejlesztői gárdájától független csoportok készítettek. Ilyen projekt az Enlightenment és az Fvwm95, ez utóbbival foglalkozunk is pár szó erejéig ezen cikk keretein belül.

Az alaprendszer mellett természetesen a modulok is jelentős fejlődésen mentek keresztül, számuk is gyarapodott. A legtöbb opcióra igaz, hogy alkalmazásonként különböző beállításokat kaphatnak, ilyesfajta óhajainkat a Style „\*” utasításokkal érvényesíthetjük. Nézzük akkor pontokba szedve a főbb különbségeket.

## Menüstílusok

A menük beállítása egyetlen helyen történik a MenuStyle utasítással:

MenuStyle feliratszín háttérszín árnyékolási\_szín fontstílus (Fvwm/mwm stílus)

Példa:

```
MenuStyle linen darkslateblue blue1 -adobe-helvetica-  
medium-o-0-*-*-16-*-*.*.*.*.*.*.* mwm
```

## Fokuszálás

Az AutoRaise funkció külön modulba került (FvwmAuto), itt már lehetőségünk van megmondani, hogy mi történjék, amikor az egérkurzor belép egy ablakba, vagy amikor elhagyja azt. Példák:

```
Module FvwmAuto 750 (alapeset)
```

```
Module FvwmAuto 750 Raise Lower (az ablak elhagyásakor  
azt legalulra helyezi)
```

```
Module FvwmAuto 750 Raise Iconify (az elhagyott ablakot  
leikonozza; meglehetősen idegesítő beállítás.)
```

További bővülést jelent a *FlipFocus* utasítás, mely a két utóljára használt ablak között váltogatja a fókuszot, továbbá a *SloppyFocus*, melynél a fókuszált ablak aktív marad, ha nem egy másik ablakba, hanem a root window fölé mozgatjuk az egeret. A 2.0.45-ös verzióban két, az ablakok elrendezését segítő új modul jelent meg, az *FvwmCascade*, melylyel a képernyőn levő ablakok szorosan egymás mögé helyezhetők úgy, hogy a fejlécük látható maradjon, és az *FvwmTile*, amely az ablakokat úgy méretezi át, hogy ne fedjék át egymást.

## Virtuális desktop beállítása

A képernyőegységekben történő méret megadása kissé változott, „DesktopSize <szám> <szám>” lett „DesktopSize <szám>x<szám>” helyett. Az egyéb beállításokat az FvwmPager modul segítségével véghezvitjük el, amit indíthatunk belépéskor, vagy menet közben menüből:

```
Module FvwmPager 0 2
```

Ebben az esetben 3 különböző munkaasztalunk lesz, persze mindegyik állhat több virtuális képernyőből.

A Pager méretét megadhatjuk fixen:

```
*FvwmPagerGeometry 116x90+3+3
```

Vagy a fizikai képernyő méretéhez viszonyítva:

```
*FvwmPagerDeskTopScale 10 10
```

Az „\*FvwmPagerFont fontnév” opció azt jelzi, hogy a virtuális képernyőn milyen fontokkal jelenjenek meg a munkaterületek nevei. Ha azt adjuk

meg, hogy „none”, akkor nem jelennek meg címkek. Ugyanígy megjelölhetők a parányi téglalapok is a Pageren belül az FvwmPagerSmallFont utasítás segítségével.

## Stílusok

A stílusok terén is jócskán kibővültek a variációs lehetőségek. Megjelent számos új funkció, emellett több lehetőségünk nyílt az alkalmazások, alkalmazáscsoportok egyedi kinézetének beállítására, amire az Fvwm1 esetén nem volt mód. Ilyenek pl. az MWM-funkciók, melyekről az Fvwm1-nél már volt szó.

Ezt a

```
Style <ablaknév> <opciók>
utasítással érhetjük el. Minden alkalmazáshoz (mondjuk „xterm”), vagy alkalmazáscsoporthoz (pl.: „*term”) hozzárendelhetünk pl. saját IconBoxot a képernyő tetszőleges területén, így csoportosíthatjuk az éppen futó programjainkat. Természetesen több alkalmazáshoz is hozzárendelhetjük ugyanazt az IconBoxot.
```

Az *ablaknév* azonosító egyébként lehet az ablak tényleges neve (ami a fejlécben megjelenik), vagy lehet az erőforrásoknál használatos azonosító is. (Pl.: az xterm esetében ez XTerm.) További újdonság a *MinIcon* – kis ikon –, mely az adott alkalmazást jelképezi.

Az ikon megjelenhet az ablakkereten valamelyik gombon, megjelenhet menüben, használhatják más modulok is, mint amilyenek az FvwmWinList, FvwmIconMan és az FvwmTaskBar.

## Fejléc, keretek

A fejléc stílusát a *TitleStyle* utasítással lehet szabályozgatni. Az első menetben a fejléc magasságát és a címszöveg igazítását állíthatjuk be. Az igazítást a következő kulcsszavakkal adhatjuk meg: „Centered” (középre igazít), „RightJustified,” (jobbra), vagy „LeftJustified” (balra). A magasságot a „height” adja meg pixelben kifejezve. Példa:

```
TitleStyle LeftJustified Height 22
```

Ha nem adunk meg paramétereket, akkor alapértelmezésben az Fvwm középre igazít, a magasságot a *WindowFont* által meghatározott magassághoz igazítja. Második menetben további hangolásra nyílik lehetőségünk:

```
TitleStyle [státusz] [stílus] [ - flag ... ]
```

Státus a következő lehet: „ActiveUp”, „ActiveDown” vagy „Inactive”, attól függően, hogy az adott stílust lenyomott, felengedett egérgombhoz vagy inaktív ablakhoz rendeljük hozzá. A „stílus” vonatkozik a fejléc színére, textúrájára, továbbá arra, hogy a fejléc az ablakkal egy síkban van („Flat”), kiemelkedik („Raised”), vagy besüllyed („Sunk”). A következő példa során az aktív ablak fejlécét úgy állítjuk be, hogy lenyomáskor lesüllyedjen és egy világosbarna-sötétbarna szingradiens jöjjön létre:

```
TitleStyle ActiveDown VGradient 64 chocolate3 #502a06
```

A következő példában olyan fejléct állítunk be, ami az ablak síkjában van, s a nyomógombok is belesimulnak ebbe a síkba:

```
TitleStyle - flat
```

```
ButtonStyle All ActiveUp (- flat) Inactive (- flat)
```

Iдетartozik még az egyszínű ablakok színbeállítására vonatkozó utasítás:

```
HighlightColor {{feliratszín háttérszín}}
```

Az ablakokat hozzá is „ragaszthatjuk” a képernyőhöz, így ha képernyőt váltunk, az ablak jön velünk. Az ehhez tartozó színbeállítást:

```
StickyColors feliratszín háttérszín
```

Az ablakkeretek stílusát nagyon hasonló módon szabhatjuk kényünkre-kedvünkre. A „stílus” ez esetben (ha megadjuk) csak „TiledPixmap” lehet. A flag lehet „HiddenHandles”, ekkor a sarkoknál sem „törik meg” az ablakkeret, ezt kiegészíthetjük a „NoInset” flaggel, amittől még tagolatlanabbá válnak ablakaink. (Mint az Fvwm95 esetében.) Példák:

```
BorderStyle Active TiledPixmap wood.xpm
```

```
BorderStyle Inactive - HiddenHandles NoInset
```

A fejlécen található nyomógombok konfigurálása is hasonló rendszer szerint történik, kiegészítve egy számmal, ami a gomb pozíciójára vonatkozik:

```
ButtonStyle gomb [státusz] [stílus] [ - flag ... ]
```

Az Fvwm1-től eltérően már nem csak „domborműveket”, hanem ikonokat is rátehetünk nyomógombjainkra:

```
ButtonStyle 6 4 50x25@1 75x75@0 25x75@0 50x25@1
```

```
ButtonStyle 4 Pixmap mini-shrink-windows-full.xpm
```

```
ButtonStyle 4 ActiveUp (Pixmap up.xpm) ActiveDown (Pixmap down.xpm)
```

Használhatjuk a „MinIcon”-t is:

```
ButtonStyle 1 MiniIcon
```

A fejléc és a keret stílusait adoptálhatjuk a „UseTitleStyle”, illetve a „UseBorderStyle” utasításokkal.

Ha használjuk az „MWMButtons” opciót, mely maximalizáláskor invertálja a gombon levő „dom-

borművet”, akkor lehetőség nyílik arra, hogy bármelyik gombhoz hozzárendeljük ezen funkciót az MWmButton fflagel.



## Menük, függvények

Az Fvwm1-től eltérően a menüket az „AddToMenu” paranccsal hozhatjuk létre, az egyes menüpontok hozzáadása a „+” kezdetű sorokban történik. Ha ezek elfognak, a menü felépítettnek tekinthető, tehát nincs külön „menüvége” szimbólum, mint a régi változatnál (End Popup). Újdonság az is, hogy a menüket meg is lehet szüntetni a „DestroyMenu” kulcsszóval, majd akár újra is lehet építeni egy új „AddToMenu”-vel, így az ablakkezelő újraindítása nélkül is változtathatunk konfigurációnkon. Következzék a példa, amelyben létrehozuk a főmenüt, hozzáadunk néhány gyakrabban használt programot, egy új sort, majd almenüket a „Popup” segítségével. A címetek körülölelő idézőjelek között „%”-ok közé beszúrva állíthatjuk be, hogy milyen ikont szeretnénk ahhoz a ponthoz rendelni.

```
AddToMenu RootMenu          "Főmenü"          Title
+ "%mini-sh.xpm%xterm"      Exec exec xterm -sb -sl \
  2048 -ls -font 7x14 -ms red -s -rw &
+ "%mini-calc.xpm%Számológép" Exec xcalc &
+ "%mini.xman.xpm%Manuál"   Exec xman &
+ ""                          Nop
+ "%mini-run.xpm%Alkalmazások" Popup Applications
+
```

```
+ "%mini.exit.xpm%Kilépés"   Popup Quit-Verify
Függvényeket hasonló módon az „AddToFunc” kulcsszóval definiálhatunk:
```

```
AddToFunc Move-or-Raise "I" Raise
+ "M" Move
+ "D" Lower
```

A függvényt hozzárendelhetjük mondjuk a bal egérgombhoz (persze hozzárendelhetünk egy billentyűzetkombinációhoz, vagy egy menüpontoz is):

```
Mouse 1 TS A Move-or-Raise
```

Tehát ha a fejlécre vagy a keretbe kattintunk az egérrel, akkor az felülre helyezi az ablakot („C”,

vagy jelen esetben egy általános kiváltó esemény, „I”), ha lenyomva tartjuk az egérgombot, akkor mozgatásra nyílik lehetőségünk („M”), kettős kattintás esetében („D”) ablakunk legalulra kerül. Egy függvényt is meg lehet szüntetni menet közben a „DestroyFunc” parancs segítségével.

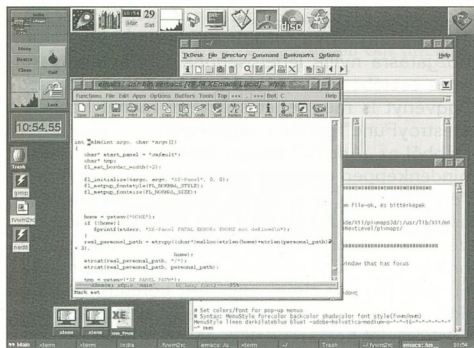
Ebből az „AddTo” kategóriából még egy funkciót mindenképpen érdemes megemlíteni, ez az *AddToDecor/DestroyDecor* páros, mellyel a különféle stílusjegyeket összefoghatjuk egy név alá, ezáltal egyszerűbben tudunk egyedi beállításokat rendelni különféle alkalmazásainkhoz. A következők példában létrehozunk egy sima, egyszerű keretet, amely ha aktiv, mutatós, fémes hatású fejléccet kap, majd ezt a dekorációt hozzárendeljük az xterm-hez:

```
AddToDecor saját
+ TitleStyle ActiveUp (VGradient 64 gray80 gray10 - flat) \
  Inactive (- flat)
+ ButtonStyle All ActiveUp (- flat) Inactive (- flat)
+ BorderStyle - HiddenHandles NoInset
+ HighlightColor white gray40
Style "saját" UseDecor saját, Color white/grey40,
HandleWidth 5
Style "xterm" UseStyle saját
```

## FvwmButtons

Az FvwmButtons modul segítségével egy nyomógombokból álló ablakot helyezhetünk el képernyőnkön. A gombokra ikonokat rakhatunk, sőt más alkalmazásokat (pl. óra, xload vagy éppen az FvwmPager) is beleintegrálhatunk. Az elődhoz, a GoodStuff modulhoz képest természetesen van néhány újdonság, talán a legemléstreméltóbb a „Container”, mellyel egy gombon belül további gombokat helyezhetünk el. A „Container” definícióját egy „\*FvwmButtons(End)” sossal zárhatjuk le. A gombok természetesen 3D hatásúak, a mélységet a „Frame” utasítással szabályozhatjuk, ha negatív számot adunk meg, akkor a gomb be fog süllyedni az ablak síkjához képest. A példában egy egyszerű modul hozunk létre egy FvwmPager-rel, két ikonnal, három kis, Containerben elhelyezett gombbal. A maradék gombon belül az xload program fog futni.

```
*FvwmButtons(2x1 Swallow "FvwmPager" "FvwmPager 0 0")
*FvwmButtons(1x1, Container(Rows 3, Columns 1, Frame 1))
  *FvwmButtons(Title Move , Action Move)
  *FvwmButtons(Title Resize , Action Resize )
  *FvwmButtons(Title Close , Action Close)
*FvwmButtons(End)
```



```
*FvwmButtons Quit bomb_3d.xpm Popup Quit-Verify
*FvwmButtons(1x1 Swallow (UseOld) XLoad 'Exec xload - \
nolabel &')
*FvwmButtons Lock lock.xpm Exec xlock -nolock -mode
random +enablesaver &
```

Mint már említettük, létezik az Fvwm2-nek egy változata, melynek kinézetét a Windows 95 kezelőfelülete ihlette meg. Ha ezt az ablakkezelőt ráadásul kombináljuk az Xaw95-tel (mely a MIT Athena widgetek 3D-s változatának egy sajátos implementációja), akkor Linuxunk megjelenése első pillantásra szinte megkülönböztethetetlenül válik a Windows95-étől. Néhány kivételtől eltekintve az Fvwm95 konfigurálása megegyezik az Fvwm2-ével, nem használhatjuk viszont a fejlécek esetében a különböző textúrákat, a keretek is csak egyszínűek lehetnek. A menükre más beállítási mód érvényes, a fontot külön, a „MenuFont

<fontnév>” utasítással állíthatjuk be. A szinkiosztás is egyszerűbb módon és kevesebb variációs lehetőséggel történik:

```
DefaultColors {menü feliratok ablak háttér fejléc fel-
iratok fejléc_háttér}}
```

Az „ablak\_háttér” itt az összes ablak háttérszínére vonatkozik, beleértve a menüket is. A „fejléc\_feliratok” és a „fejléc\_háttér” szín az inaktív ablakok esetében érvényes. Az aktív és a sticky (képernyőhöz ragasztott) ablakok színét a korábban leírtak szerint állíthatjuk be.

Az Fvwm95-tel jelent meg a TaskBar nevű modul, mely elég élethű koppintása a Win95 Taskbar-jának. Ez egy olyan speciális ablak, mely az éppen aktuális ablakok listáját tartalmazza, az aktív ablak gombja benyomott állapotban, kissé világosabb színben jelenik meg. Ha valamelyik gombra rákattintunk, akkor az ahhoz tartozó ablak válik aktívvá, ha ikonba volt csukva, ki is nyílik. A TaskBar tartalmaz még egy „Start” gombot (természetesen a feliratot és az idetartozó kis ábrát lecserelehetjük másra), ez egy külön definiált menüt hoz elő, valamint található rajta egy kis óra, és egy levelet ábrázoló ikon, melyre kattintva kedvenc levelezőprogramunkat hozhatjuk elő. Itt jegyezném meg, hogy a 2.0.45-ös verziótól kezdve az Fvwm2-ben is használható már a TaskBar. Ehhez csak annyit kell tennünk, hogy a ModulePath-nál be kell állítanunk azt az alkönyvtárat, ahol a TaskBar található (általában /usr/lib/X11/Fvwm95-2/). Egyébként a megjelenést tekintve is szinte egy az egyben „emulálható” az Fvwm95 az Fvwm2-n belül is, sőt menet közben pl. az AddToDecor/DestroyDecor segítségével váltogathatunk is mondjuk Win95- és Motif-stílus között. ■



Kégl Tamás  
(monty@python  
.mars.vein.hu)

# AfterStep

Az AfterStep részben a BowMan ablakkezelő utódjának tekinthető, melynek alkotója Bo Yang volt. A projekt célja az volt, hogy emulálják a NeXtStep operációs rendszer kezelői felületét, kiegészítve az X-Window rendszer és az Fvwm ablakkezelő előnyös tulajdonságaival.

Talán nem felesleges megemlíteni, hogy a NeXtStep egy BSD-alapú Unix, melynek a grafikus felülete a legtöbb Unixétól eltérően nem az X-Window, hanem egy attól lényegesen eltérő PostScript-alapú grafikus rendszer. Később aztán megváltoztatták a projekt nevét a jelenlegire.

Az AfterStep egy nagyon tetszetős ablakkezelő, azonban hátrányaként meg kell említenünk az aránylag nagy memóriagigényt. Ez persze elsősorban az összetett és szingazdag beállításoknál léphet fel. További problémát jelenthet, hogy 256 színű videomód használata esetén nem igazán érvényesülnek a színátmenetek, ráadásul a színpaletta is meglehetősen gyorsan betelhet, tehát inkább a 16 bites, vagy afeletti színmélység ajánlott.

Az AfterStep konfigurációs parancsai nagyrészt megegyeznek az Fvwm 1.24 parancsaival. A különbségek elsősorban abból fakadnak, hogy itt már lehetőség nyílik textúrák alkalmazására az ablakkeretek és a menük esetében.

Az ikonba csukott ablakok is máshogy néznek ki, mint az Fvwm esetében, egyenlő méretű háttér tartozik mindegyikhez, az ablakok neve kis fejléccen helyezkedhet el. Jelentős újdonság a Wharf modul, amely a GoodStuff panel továbbfejlesztésének fogható fel.

Zsúfolt desktopoknál hasznos lehet egy jól irányzott *Shade* parancs, mely az ablakból csak a fejléccet hagyja meg, a többi részét átmenetileg eltünteti. (Ez a funkció egyébként tudtommal a MacOS-ben jelent meg először, de már az Fvwm2-ben is megtalálható.) Alapértelmezésben a „Shade” kettős kattintással hozható elő. A

jobb egérgombbal maximálisra növelhetjük ablakunk méretét, a bal felső sarokban levő gombra kattintva ikonba csukhatjuk az ablakot, míg a jobb felső sarokban levő „X” gombbal bezárhatjuk azt.

## Textúrák

Az ablakkeretek és menük esetében lehetőségünk nyílik színátmenetes textúrák alkalmazására. Ez azt jelenti, hogy definiálhatunk egy kezdeti, meg egy végső színt, melyek a fejléc vagy a menü ellentétes pontjain találhatóak, ezek lesznek a színátmenet szélső értékei. A textúra típusának beállítása a *TextureTypes* paranccsal történik:

*TextureTypes* fejléc fókuszálatlan\_fejléc sticky\_fejléc menüpontok menü cím

Lehetséges értékek:

- \* 0: nincsen textúra;
- \* 1: Wharf-stílusú textúra; a bal felső és a jobb alsó pontok között;
- \* 2: függőlegesen változó szín, a szélső pontok legalul és legfelül vannak;
- \* 3: függőlegesen változó szín, az egyik szélső szín közepén, a másik alul és felül található;
- \* 4: ugyanaz, mint a „2”-es eset, vízszintes színátmenettel;
- \* 5: ugyanaz, mint a „3”-as eset, vízszintes színátmenettel.

Példa:

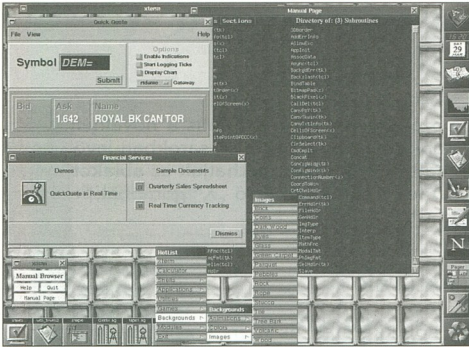
*TextureTypes* 1 2 1 4 3

A *TextureMaxColors* paranccsal megszabhatjuk, hogy a színátmenetek hány színből álljanak. 16 bites színmélység esetén ez lehet akár 64 vagy 128 is (ha nem adjuk meg, akkor alapértelmezésben 128 szokott lenni), 8 bit esetén érdemesebb ezt kisebbre választani:

*TextureMaxColors* 10 10 10 10 10

A színátmenetek definiálása egyszerűen a szélső színek megadásával történik:

```
TitleTextureColor #3030d0 #101040 (aktív fejléc)
UTitleTextureColor #a0a2d1 #505261 (inaktív fejléc)
STitleTextureColor #a0f2c1 #304261 ("ragadós" fejléc)
MTitleTextureColor red black (menü cím)
MenuTextureColor #703070 #d0d0d0 (menüpontok)
```



## Wharf

Az AfterStep jellegzetes eszközpánelleje a Wharf, mely elődjénél, a GoodStuffnál lényegesen látványosabb. Lehetővé teszi, hogy az egyes gombok harmonikaszerűen becsúszzanak a legfelső gomb alá (ezt a középső egérgombbal érhetjük el), a gombokból ugyanakkor vízszintesen további sor nyomógombok csúsztathatók elő. (Erre a bal egérgomb szolgál.)

Természetesen arra is van mód, hogy bizonyos alkalmazásokat (xload, Pager) passzírozzunk be egy gomb alá. A CD-mellékleten megtalálható egy mintaállomány (.steprc), benne természetesen a Wharfra vonatkozó beállításokkal, most csak néhány kevésbé egyértelmű pontra térnénk ki:

- *\*WharfGeometry -0+0*: alapértelmezésben a Wharf a jobb oldalon található, ha +0+0 értéket állítunk be, akkor áthelyezhetjük a bal oldalra.
- *\*WharfTextureType 1* alapértelmezés szerint a szingradient úgy állítja be, hogy a színátmenet a bal felső saroktól a jobb alsó sarokig tart.

Ekkor használhatjuk a *\*WharfTextureColor* parancsot, pl.:

- *\*WharfTextureColor #ff3030 #000010*: ha 128-at adunk meg TextureType-nak, akkor xpm formátumú képet tehetünk textúráként a gombokra:

- *\*WharfTextureType 128*

- *\*WharfPixmap walnut.xpm*

## Hangosítás

Az AfterStephez tartozik hangmodul is, neve „Audio”. A következő rövid példában a virtuális képernyőváltás eseményéhez rendeljük hozzá egy kakukkosóra hangját, ehhez a wavplay audioplayert használjuk.

Module Audio

```
*AudioPlayCmd /local/bin/wavplay 2>/dev/null
```

```
*AudioDir /local/include/sounds
```

```
*Audio iconify cuckoo.wav
```

Itt említeném meg, hogy az Fvwm esetében gyakorlatilag ugyanezekkel a parancsokkal történik az audiomodul konfigurálása, csak ott az „Audio” helyett mindenütt az „FvwmAudio” elnevezés használatos. További érdekesség, hogy a 2.0.45-ös verziótól kezdve az fvwm2-ben is használhatjuk a Wharfort, sajnos egyelőre bizonyos megszorításokkal. (A textúrával gondjaink lehetnek, továbbá nem működik a Swallow.)

## Végezetül

Lehetséges, hogy első olvasatra kissé bonyolultnak tűnnek az itt leírtak, de néhány példát megtekintve kellő számú próbálgatással rendkívül kényelmes munkakörnyezetet hozhatunk létre, hála az X-Window grafikus rendszer végtelen rugalmasságának. ■

Tulassay Zsolt  
(tzsolt@podolin  
.piar.hu)

# Motif, CDE, AcceleratedX

Aki ismeri a különböző Unixok grafikus kezelőfelületét, tudja jól, mennyire eltérhetnek azok egyes esetekben egymástól. Ez pedig nemcsak a sokszor különböző kezelési problémáját veti fel, de megnehezíti az egyes alkalmazások átvitelét is egyik rendszerről a másikra.

A szabványosítás szükségességét felismerve született meg a Common Desktop Environment, azaz az Egységes Munkakörnyezet.

## A CDE elemei

Mi is valójában a CDE? Tulajdonképpen egy Motif-ra épülő, gyártófüggetlen felhasználóifelület-szabvány, kiegészítve jó néhány hasznos felhasználói programmal. Ma már elérhető szinte az összes létező Unix platformra, egységes felhasználói, fejlesztői



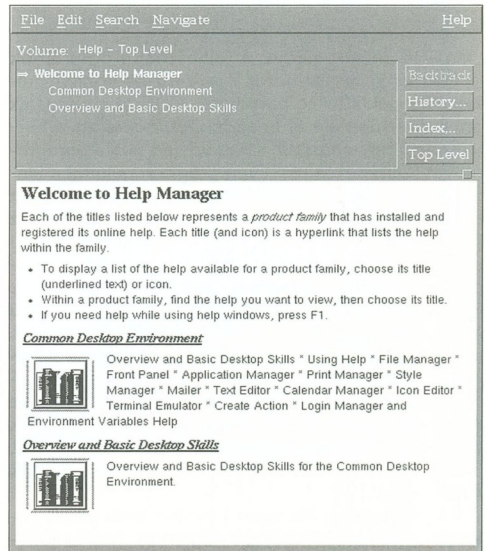
A Style Manager

és programkörnyezetet biztosítva. (Része például az HP-UX-nek, az IBM AIX-nek, a Digital Unixnak, az újabb Solarisoknak stb.)

A Motif az Open Software Foundation által megalkotott szabványos elemgyűjteményből (elem = widget; ez határozza meg az ablakokon belül a gombok, menük, scrollbarok stb. alakját, méretét, helyét, valamint minden egyéb, megjelenését illető tulajdonságot) és az mwm (Motif Window Manager) nevű ablakkezelőből áll. Ez utóbbi végzi az abla-

koknak és azok kereteinek, valamint például az egyes desktopoknak, ikontrárainak és gyorsmenüknek a megjelenítését, kezelését.

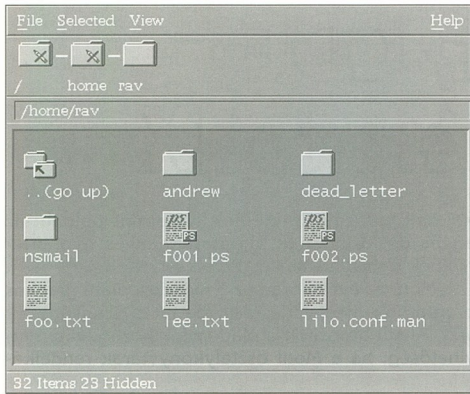
A Motif az évek során az X-Window alatti grafikus programok megjelenésének de facto szabványává vált. Általa biztosított a platformok közti könnyű átvihetőség: ha a Motif osztálykönyvtárak rendelke-



Help Manager: a CDE súgója

zésre állnak, általában nem igényel túl nagy munkát egy alkalmazás átvitele más platformra.

Meg kell jegyezni, hogy a Common Desktop Environment a CDE Motifra épül, ami némileg különbözik a „hivatalos” OSF-féle Motif szabványtól. A CDE Motif legtöbb új funkcióját azonban beemelték a Motif 2.0-ba.



Így néz ki a file-kezelő

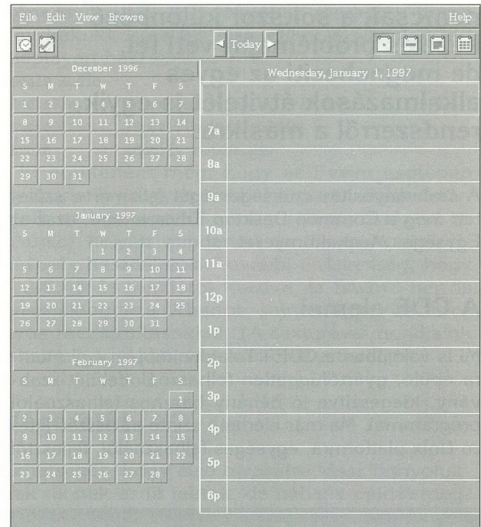
A CDE másik eleme, a felhasználói programok gyűjteménye biztosítja az egyszerű és könnyű kezelhetőséget. Elvileg bármilyen, CDE-re épülő Unix grafikus felület elé ül le az ember, ugyanazt a környezetet, ugyanazokat a kezelő- és beállítóprogramokat kell(ene) látnia. Ilyen program például a file-kezelő, a ságó, a Style Manager (a grafikus felület beállítómodulja), a grafikus levelezőprogram, az ikonszerkesztő, a Print Manager, az xdm bejelentkező modulja, egyes rendszerkezelő szoftverek, valamint egy – nagyon hasznos – ikon-tár, a Front Panel. Ezt leginkább a Goodstuff, a Pager és a root menü kiterjesztett, grafikus változatának lehetne nevezni. Előnye, hogy teljes mértékben támogatja a drag-and-drop-ot („húzd és ejtsd”), így nagyon egyszerűen helyezhetünk el rajta újabb ikonokat, illetve nyomtathatunk ki például egy file-t. Használhatjuk az objektumok beágyazását, illetve linkelését, ahogy azt már a Microsoft féle OLE-vel megszoktuk (már aki :-).

A CDE Session Manager lehetővé teszi eltérő jellemzőjű (képernyőméret és -felbontás, kapcsolódási sávszélesség stb.) sessionok definiálását, ahol a DISPLAY környezeti változótól függően mindig a megfelelő indul el.

Tartalmaz a CDE fejlesztői programokat is, mint például az Application Buildert, ami ugyan csak a programok grafikus keretét készíti el, de ezek után már igazán nem nehéz beilleszteni a valódi kódot.

Ez így mind nagyon szép, de... A bökkenő ott van, hogy a tulajdonos Open Software Foundation a Motif forráskódját nem tette ingyenesen elérhető-

vé, ahogy azt linuxos-unixos programok esetében szinte elvárni az ember. Éppen ezért egy ideig nem is volt elérhető Linux alatt. Ez azonban időközben megváltozott: ma már a CDE linuxos változatát készítő Xi Graphics mellett a RedHat és az InfoMagic is kínál elérhető áron Motifot Linux alá. Sőt, linuxos buherátorok egy lelkes csoportja elkészítette annak ingyenes, a GNU licenc alá eső változatát, a *lesstif*-et is. Ez kompatibilis a Motif 1.2-es implementációjával, biztosítva a megfelelő környezetet a Motifot igénylő programok számára. (Viszont az időközben elkészült Motif 2.0-t nem támogatja.) A legtöbb



A Calendar, most éppen havi bontásban

ilyen programnak azonban általában rendelkezésre áll statikusan fordított, külső osztott libraryket nem igénylő verziója is.

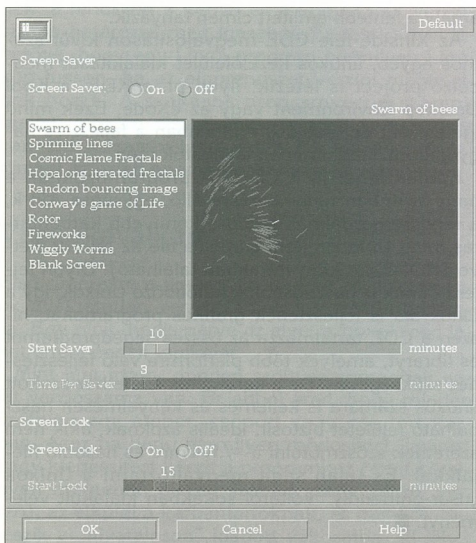
## A CDE Linux alatt

A Common Desktop Environment sem létezett jó ideig Linux alá. Csak a múlt év végére készült el az amerikai Xi Graphics (<http://www.xig.com>, korábbi nevén XInside) a linuxos változattal. Sajnos ez sem ingyenes, sőt elég borsos az ára. Hardverigénye



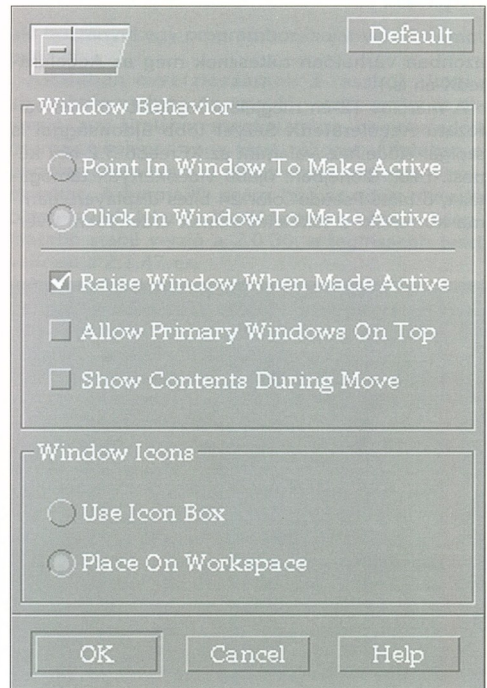
Egyszerűen lehet hátteret megadni

(min. 16 Mbyte RAM, min. 80 Mbyte lemezterület) ma már nem számít nagyon soknak; ezen kívül csupán a – véletlenül szintén az Xi Graphics által készített – AcceleratedX X-szerverre van szüksége a működéshez (erről lásd alább). Cserébe kapunk egy minden igényt kielégítő grafikus munkakörnye-



A képernyőkímélő beállításai

zetet, plusz az említett, jól használható segédprogramokat, valamint egy teljes Motif 1.2 fejlesztői környezetet. A 97-es UniForumon, március 12-én jelentette be a cég a CDE Business Desktopot, amely egy komplett Linux-, avagy FreeBSD-alapú hálózatos operációs rendszer. Ez az előzőekhez ké-



Az ablakok jellemzői

pest számos új elemet, illetve továbbfejlesztést tartalmaz: már az új AcceleratedX-et, egy „nemzetközi csomagot”, amely számos európai nyelvet (és a japánt is) támogat, valamint továbbfejlesztett drag-and-dropot a grafikus felületen stb.

## Az AcceleratedX szerver

Az AcceleratedX elsősorban gyorsaságában (meg persze árában) különbözik az XFree-től (a 3.1.2-t

alapul véve). Érdemes kipróbálni: xteddy-t mozgatni egy xterm-ablak felett XFree 3.1.2 és AcceleratedX esetén. (Az XFree86 3.2 és a még újabb béták

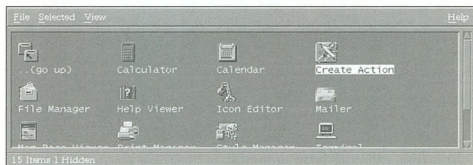


A Front Panel

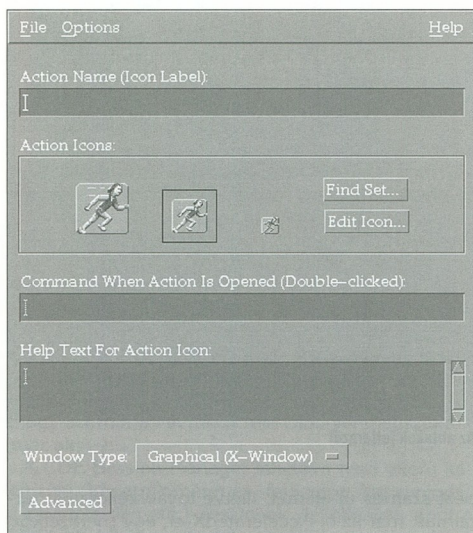
azonban várhatóan túlesznek még az AcceleratedX-en is.)

A március 12-én megjelent legújabb, 3.1-es változatú AcceleratedX Server több újdonsággal is szolgált elődeihez, valamint az XFree86 3.2-höz képest. Ezek dióhéjban: újabb videokártyák támogatása, 8 bites PseudoColor 24 bites displayen, gamma-korrektció a tökéletes színek érdekében, over-

és MultiHead) több Unix alá kapható X-szerver másik előnye, hogy immáron több mint 450-féle videokártyát (124 különböző chipsetet) támogat, köztük a Matrox, Diamond NeoMagic és Number Nine, továbbá az S3 Virge kártyákat is. Ez talán az egyik fő ok, amiért a Caldera is úgy döntött, hogy leendő OpenLinuxát az Xi Graphics AcceleratedX-ével szállítja. Az X-szerver konfigurálása sem különö-



A fontosabb alkalmazások mappája



Így hozhatunk létre új objektumokat

layek használata, 2048×2048-as virtuális desktop és megnövelt grafikus teljesítmény (1005117 Xstones, illetve 24 Xmarks).

A három változatban (Desktop, Laptop/Portable

sebben bonyolult, hála a menüs beállítóprogramnak. Létezik ráadásul egy mini-HOWTO (az XFree86-XInside) is, amely részletesen leírja, hogyan konvertáljuk jól bevált XFree-s Modeline-beállításainkat az új formátumra. Vállalkozó szelleműek akár ki is próbálhatják: az AcceleratedX demó verziója ugyanis letölthető a cég WWW-szerveréről, amely a fentebb említett címen tanyázik.

Az XInside-féle CDE-megvalósításon kívül számos egyéb, linuxos kezelőfelület kialakítását megcélzó project is létezik: ilyen pl. a KDE, a Kool Desktop Environment vagy az eXode. Ezek mind ígéretek, de – sajnos – általában a fejlesztés nagyon korai stádiumában találhatók még.

Kiknek lehet jó a CDE? A programozóknak a nagy számban rendelkezésre álló, egységes API-k, a rendszergazdáknak pedig a könnyebb karbantartás miatt (minden konfigurációs file az /etc/dt illetve \$HOME/.dt könyvtárakban található). Olyan fejlesztőknek is hasznos, akik különböző Unixok, így a Linux alá is fejlesztenek grafikus programokat, hiszen a CDE tartalmazza az összes segédprogramot és libraryt, amelyek több platformra való fejlesztéshez szükségesek. De a karakteres felülethez szokott linuxos számára is kellemes és kényelmesen használható felületet biztosít. Ideális azoknak, akik nem szeretnek szöszmötölni a ~/.fvwmrc és hasonló filejaikkal. És talán segít megváltoztatni a Unixok „user-unfriendly” voltáról sokakban kialakult véleményét. ■

Czakó Krisztián  
(slapic@fido.hu)

# Hol lelhetünk Linuxot?

**A Linux operációs rendszer alapvetően a kernelből és a működéséhez szükséges könyvtárakból (librarykból) áll. Minden egyéb része csak kiegészítő, különálló szoftver (ezért van mindennek saját verziószáma).**

Amikor egy Linuxot telepítünk, valamilyen tesztést (disztribúciót) teszünk fel, azaz egy olyan rendszert, ami az alapokból és válogatott programokból áll. A létező disztribúciók többségében nagyon hasonló programok vannak, hiszen mindenhol szükség van shellre, editorra stb., tehát a disztribúciók között alapvető különbségek nincsenek, a rendszert a kernel és a könyvtárak verziószáma határozza meg igazából. Ez teszi azt is lehetővé, hogy egy Linux-program bármilyen Linuxon fusson, ha az alaprendszer verziói megfelelnek hozzá. A szokásos programverzió-számok mellett minden disztribúciónak saját verziószáma van. Ezt a számot lehet leginkább hasonlónak tekinteni az üzleti operációs rendszerek verzióival.

## Verziókavalkád

Első ránézésre a linuxos szoftverek és disztribúciók átláthatatlan verziószámok káoszát tartalmazza. Egy picit megismerve az elvet azonnal meg lehet érteni ennek logikáját.

Mint fent írtam, minden szoftvercsomagnak saját verziószáma van. Ez abból ered, hogy minden szoftvert egy ember, csoport vagy cég tart karban. Azaz ha bárki bármilyen módosítást, javítást vagy kiegészítést készít a programhoz (amit könyvnyen megtehet, hiszen szabad szoftvekről van szó), az elküldi a karbantartóknak, és majd ők beteszik a „hivatalosan” kiadott változatba. El lehet képzelni, milyen káosz lenne abból, ha mindenki,

aki hozzányúl egy programhoz, saját verziószámot adna ki...

Általában a verziószámok 3 részből állnak, *xx.yy.zz* formában. Az *xx* a fő verzió, az *yy* az alverzió, ami általában meghatározza, hogy az adott csomag végleges vagy teszt változat. Általában a páratlan szám teszt, a páros szám végleges verziót takar. A harmadik szám (*zz*) az adott alverzió patch-levelét jelenti. Jó példa erre a kernel. A legfrissebb stabil verzió a 2.0.30, a legfrissebb teszt változat a 2.1.47-es.

A disztribúciók verziója ettől eltérő, az az adott terjesztést kibocsátó csoport egyedi számozása, ott nem érvényesek a fentiek. A disztribúciók verziója azonban meghatározza a többi program verzióját oly módon, hogy a disztribúcióban mindig adott verziójú csomagok vannak. Persze, amikor kiadnak hibajavításokat, ez felborul, de azért még jól követhető, hogy épp milyen programokat tartalmaz egy-egy disztribúció.

Manapság egyre több disztribúció érhető el. Pár évvel ezelőtt még majdnem mindenki a Slackware-t használta, ma már igen elterjedtek a Red Hat, Debian és SuSE disztribúciók, illetve a Caldera üzleti Linuxa, az OpenLinux. Ezek egy része teljesen ingyenes, egy része csak részben, és van teljesen fizetős változat is. Természetesen az alap Linux akkor is ingyenes, ha az adott disztribúcióért fizetnünk kell. Amiért pénzt szoktak kérni, azok az extra, fizetős programok, a nyomtatott dokumentáció és a támogatás (support).

## Slackware

Régi linuxosok sosem fogják elfelejteni az első Slackware-üket, de még ma is sokan használják. Eredetileg egy nagyon leegyszerűsített, DOS-ról áttérőknek kézreállóbb disztribúció volt, ma már egyre inkább a szabványokat követi. Sajnos a disztribúciót nem tartják kellő szinten karban, azaz sokszor későn jelennek meg a hibajavítások hozzá.

## RedHat

Talán az egyik legelső valamelyest üzletiesített disztribúció. Az első igazán jól használható verziója a 2.1-es volt. Ma már a 4.2-es a legújabb, a 4.1-essel a CHIP Magazin 1997. márciusi számában az olvasók is találkozhattak.

A RedHat által összerakott disztribúció alapvetően ingyenes, de a boltban kapható változatában van pár olyan program, ami az ingyenesből hiányzik, illetve adnak hozzá egy jó dokumentációt és supportot is. A RedHat cég odafigyel a biztonsági és egyéb hibákra, a disztribúció saját csomagformátumában (rpm) szinte a hiba felfedezésével egy időben jelennek meg a javítások.

Itt érdemes szót ejteni a csomagmanagerről, az rpm-ről is. A RedHat volt az első olyan disztribúció, amiben jól átgondolt csomagkarbantartó program volt, ami képes kezelni a függőségeket, installálni, frissíteni és törölni a csomagokat.



## Debian

A Debian disztribúció talán az egyetlen teljesen szabad Linux-terjesztés, amit a neve (Debian GNU/Linux) is takar. Azaz az alapdisztribúcióban csak szabad programokat találunk. Természetesen ehhez is megvannak a valamilyen korlátozást tartalmazó szoftverek (a korlátozás nem feltétlen működési vagy használati, hanem lehet terjesztési is), külön könyvtárban. Kiadványunk CD-jére felkerült minden, amit csak feltehettünk.

A csomagkezelője ennek is jól átgondolt rendszerű, a függőségek kezelésében jobb, mint a RedHaté. Biztonság szempontjából is jobb, mint a RedHat, azaz itt is azonnal jelennek meg a javítások, valamint az alapcsomagban több olyan biztonsági felügyelő szoftver van, ami a többieknél nincs meg.

A disztribúció mögött egy non-profit szervezet áll, azaz a munkát nem anyagi érdekből végzik, ami nagy előny a többiekkel szemben.

## S.u.S.E.

Ez egy német terjesztésű Linux, talán a legtöbb programot tartalmazó. Mivel német, a teljes telepítés és karbantartás az angol nyelv mellett németül is elvégezhető. Eredetileg a Slackware primitív csomagkezelésére épült, azt egészítették ki, de a legújabb 5.0-s változat már a RedHat rpm csomagkezelőjét használja, azaz a csomagkezelésről és biztonságról írottak nagyjából azonosak a RedHatéval.

## Caldera OpenLinux

Az első és ismereteink szerint eddig egyetlen teljesen üzletiesített Linux. Bár nemrég megjelent az *OpenLinux Lite*, ami ingyenesen elérhető, de ez inkább egy demó, mint komplett és jól használható disztribúció.

A Caldera a RedHat Linuxra építette rendszerét, azaz ők is az rpm csomagkezelőt használják. Ez a disztribúció tartalmazza a legtöbb fizetős programot, köztük egy ikonos grafikus felület is. (A többi disztribúcióhoz is vásárolható ilyen felület, illetve már készül a KDE, egy nagyon ígéretes, ingyenes felület.) Továbbá része minden OpenLinuxnak a Netscape Navigator legújabb változata is, emellett a Caldera tudhatja magáénak a legtöbb, Linuxra elérhető üzleti terméket.

A Lite mellett további három verziót ad ki a Caldera: a *Base* a legolcsóbb, ebben csak néhány alapszoftver található (a grafikus felület, Netscape Navigator, Crisp Lite, StarOffice noncommercial, stb.). A *Standard* változatban már a StarOffice commercial változat, Netscape Fastrack Server, Netware-kliens is található, valamint adják hozzá az Adabas-D adatbázisszerver personal változatát is. Az összességében megjelenő *DeLux* változatban a fentiek mellett komplett Netware-szolgáltatás (beleértve az NDS-t is) lesz.

A Caldera a fentiek mellett jó dokumentációt ad, jól képzett nemzetközi terjesztői és support hálózatot is üzemeltet.

\* \* \*

A fentiek mellett sok más, kevésbé ismert disztribúció is létezik, hiszen bárki bármikor csinálhat sajátot. ■



Czanik Péter  
(czanik@fang  
.fa.gau.hu)

# A SuSE disztribúció

**A SuSE-t négy diák alapította 1992-ben Németországban, akik eredetileg szoftverfejlesztéssel foglalkoztak. A Linuxot disszertációik megírásához használták, mert így otthon is Unixot használhattak, az egyetemi gépekhez hasonlóan.**

A cikk írásának idején az aktuális változat a 4.4.1. Ez tulajdonképpen a 4.4-es aktualizálása a mostani állapotokra. Így bár a doboz még 2.0.28-as kernelt emleget, de a lemezeken már újabb szerepel, s hozzá már egy „out of the box” működőképes DOS-emu is jár. A SuSE cég jóvoltából én a „nemzetközi”, azaz angol változattal ismerkedhettem meg közelebről, így ami ezután következik, azt az ezzel kapcsolatos élményeim alapján írom.

Története kezdetén alkotásuk tulajdonképpen nem is volt önálló disztribúciónak tekinthető. Átvették a Slackware aktuális változatát, irtak hozzá egy német nyelvű könyvecskét az installáláshoz, és a könnyebb konfigurálhatóság kedvéért kifejlesztették a YaST-ot (Yet another Setup Tool = Egy újabb Beállítóprogram). A program népszerűsége egyre nőtt, s lassan a teljes német nyelvterületen kapható volt összeállításuk. 1995 vége gyökeres fordulatot hozott. Szakitottak a hagyományokkal, és saját összeállítással jelentek meg a piacon. Ennek csak egyik oka volt, hogy a Slackware fejlesztése akadozott, és sok program igencsak régi és bugos volt benne. A Slackware-ből hiányzott a fejlett runlevel rendszer. A Linux fejlődése egyre inkább felgyorsult, s egy másik fejlesztés átvétele, tesztelése, fordítása már annyira késleltette volna a programok megjelenését, hogy már megjelenéskor elavultak lettek volna. Így külső fejlesztők bevonásával elszakadtak „ősüktől”, s az önállóság útjára léptek.

Várható volt, hogy a legújabb cél a nemzetközi elismertség megszerzése lett. A cégnek nemrég jött létre amerikai leányvállalata, melyben az eredeti négy helyett most már több mint harminc alkalmazott dolgozik. Az új változatokat már nemcsak németül, hanem angolul is elkészítik. Ennek következtében már angolul is olvasható a felhasználói kézikönyv és működik a YaST. A gyakran feltett kérdésekre választ adó „support adatbázis” egyelőre csak németül olvasható a WWW-n, de készül az angol változat is. A kézikönyv egyébként a legtöbb itt felmerülő kérdésre választ ad.

## A disztribúció

Itt most természetesen az következik, hogy miért tartom a legjobb Linux-változatnak ezt az összeállítást. Igyeksem azért a vallásos rajongás helyett a földön járva ismertetni eddigi tapasztalataimat. Bár ez igen nehéz feladat, de megpróbálok. Eladási statisztikáik szerint egyébként rajtam kívül még kb. 100 000 hívót sikerült idáig gyűjteniük. Először is fontos megemlíteni, hogy milyen médiákon érhető el a disztribúció. Mint legtöbb társa, ez is két változatban érhető el. Kapható szép dobozban CD-lemezekben, illetve letölthető a Hálózatról ftp-vel. Ez utóbbi nem tartalmaz néhány CD-n megtalálható programot, illetve a live-filesystemet, de ez csöppet sem ront használhatóságán. A doboz két fontos dolgot tartalmaz. Az egyik egy részletes felhasználói kézikönyv, a másik egy három CD-t tartalmazó tok. Eből kettő használható az installáláshoz, a harmadik lemez pedig a live-filesystemet tartalmazza.

Az első lemezen főleg az alapinstallációhoz szükséges file-ok találhatók meg. Ezek a különféle hardver-összeállításokhoz előkészített root- és boot-lemezek, valamint található rajta néhány DOS-os programocská is, köztük a fips, az adatvesztés nélküli particiódaraboláshoz, a loadlin a dos alatti bootoláshoz, egy NFS-kliens, és még néhány hasznos és kevésbé hasznos segédprogram.

Szintén itt olvasható a teljes német és angol nyelvű dokumentáció, a kézikönyv .ps és .dvi formában, angol és német FAQ-k, HOWTO-k. A hátra maradt helyet a csomagok töltik ki. Ezekből izelítő-

ül néhány a Linux-listán népszerű, vagy nekem tetsző program: XWPE, Scilab, Xpenguin, Povray, Xfractint, SNNS, Lyx, jazz, MuPAD, Postgres95 stb.

A második lemezen található meg a teljes forráskódot, az „egyéb” kategóriába tartozó programokat, mint például asztrológia, vonalkód-leolvasó és hasonló jól besorolható dolgokat. Itt vannak a pénzes szoftverek demói, köztük néhány kevésbé lebutított is, melyek miatt nem lehet a CD-k teljes tartalmát kirakni ftp-re.

Ezenkívül megtalálható még rajta a StarOffice Beta3 változata. Egy régi CHIP CD-ről könnyen legyűjthető a szükséges libraryk, ezek után már könnyedén felinstallálható a program. Kellemes meglepetésként csak hosszas kísérletezés eredményeképp sikerült hibát okozni a programban. Ami legjobban meglepett, az a WinWord import szűrő volt, mely a gyorsmentett magyar szöveget is hiba nélkül olvasta.

## Az „élő” rendszer

Az utolsó lemezen a live-file-system van. Ez azért jelentős „készítmény”, mert ha csak ismerkedni szeretnénk a Linuxszal és a hozzá tartozó csomagokkal, akkor elég csak 30 megányi anyagot felrakni a gépre, a maradék 600 mega a CD-ről jön. Így kipróbálhattam jó néhány programot, melyeket HDD-m véges volta miatt nem állt volna módomban.

Néhány érdekesség: hirtelenjében hét window manager között válogathattam, és mindegyik elsőre működött rendesen CD-ről, még az OpenLook is. Működött a dosem is, természetesen csak a -A opcióval (azaz lemezről), de egy fdisk /mbr hatására ezt a problémát is kiküszöbölhettem.

## Igazi dokumentáció

A dobozban található 450 oldalas angol, illetve német nyelvű könyv nagyon kellemesen használható. Ez tényleg olyan, hogy ha valaki olvassa, és közben dolgozik a Linuxán, akkor a könyv alapján megismerkedhet az alapokkal, fontosabb fogalmakkal, és „minden úgy működik, ahogy a nagykönyvben meg van írva”.

A könyv elején egy részletes installálási tanácsadót olvashatunk. Ehhez kapcsolódik a függelék több része, amely részletesen leírja a támogatott hardvereket. A következőkben a kernelnek átadható paramétereket, majd a YaST-ot, a SuSE által fej-

lesztett istalláló, konfiguráló programot ismerhetjük meg részletesebben. Ezt LILO leírás és SuSE FAQ követi. A Linux-gyorstalpalóból megtudhatjuk az alapvető file-műveleteket, a man, a ps és mount használatát, és természetesen a kedves szövegszerkesztő, a vi alapjait.

Egy teljes külön fejezetet szán a könyv az Emacs használatának ismertetésére, majd egy következő a kernelfordítás rejtelseinek. Az utolsó fejezetek az X-felületet, a hálózatot, az emulátorokat és a csomagleírásokat taglalják részletesen. A függelékben megismerhetjük a támogatott hardware eszközöket, a nyomtató beüzemelésének módját, a könyvtárstruktúrárt és az installációs lemez készítésének menetét.

## Telepítünk

Az installálás és konfigurálás a YaST nevű program segítségével végezhető. Ez csak installáláskor nélkülözhetetlen, mert szinte minden funkciója megoldható parancssorból is, illetve a /etc/rc.config nevű file segítségével. Ez alól talán csak a live-file-system ki/be kapcsolása a kivétel. Segítségével lehet .rpm csomagokat is installálni, .deb-et a cikk írásakor sajnos még nem. Ha nem csak a CD-n levő programokat szeretnénk használni, akkor rendelkezésünkre áll egy linkgyűjtemény, amelyek segítségével javítják a kompatibilitást. X-felületként az XFree 3.2 található meg a SuSE-ban. Ez már tartalmazza az S3 Virge és a Matrox Millennium támogatását, sok egyéb új kártya mellett. Sokkal gyorsabb, mint elődje, s ez nemcsak a futásra, hanem az indulására is vonatkozik. Ez annak köszönhető, hogy a window managerek nem generálják mindig újra konfigurációs file-jaikat, csak új csomag feltételek, vagy ha a felhasználó kézzel elindít hozzá egy programot.

Lehetőség van német nyelvű menürendszer felépítésére is. Mindenki ízlése szerint választhat magának window managert, hétféle program állítja nehez döntés elé a felhasználót: AfterStep, BowMan, ctwm, fvw1, fvw2, fvw3, olw3m. Ezeket egészíti ki néhány funkcióval a SuSEwm-nek keresztelt téglalapocská a képernyőn, amellyel váltogatni lehet a virtuális képernyőkön, valamint a pontos idő, dátum, coolmail, és a leggyakrabban használt programok ikonjai is rajta vannak.

A SuSE-t mint alapvetően CD-n megjelenő rendszert, főleg új kiadáskor frissítik fel. Ez természetesen nem jelenti azt, hogy a közte eltelt időben nem

történi semmi, de ilyenkor főleg csak a menet közben kiderülő egy-két hibát korrigálják. Ezek a frissítések az <ftp://ftp.suse.com> címről tölthetők le. Átvehetünk csomagokat a Florian La Roche-féle SuSE/Jurix-ból is, az esetek nagy többségében ez nem okoz problémát, mert ő is részt vesz a CD fejlesztésében. A SuSE rendszert nagyfokú biztonsága és nagyon széles skálájú programellátottsága miatt tudom ajánlani szerver gépre és otthoni felhasználásra egyaránt.

## Egy tipikus installálás története

Többféle médium áll rendelkezésünkre installáláshoz. Legtöbb esetben természetesen a CD-ket használja az ember forrásként, de lehetőség van erre hálózaton át (NFS), illetve tetszőleges előre felmontolt könyvtárból. Ez utóbbi esetben DOS-on kívül VFAT, EXT2FS és az NFS is rendelkezésünkre áll. Miután a legfőbb installációs média a CD-ROM, ezért a továbbiakban erről lesz szó.

Első feladatunk, hogy gépünket bebootoljuk a Linuxszal. Ha még van DOS a gépen, akkor ezt leg-egyszerűbben úgy tehetjük meg, hogy átváltunk az F1: meghajtóra, vagy ahol az *Installation 1.* feliratú CD található, és beírjuk, hogy Setup. Ekkor a program néhány, a hardverünkre utaló keresztkérdés után elindítja a megfelelő kernelt a CD-ről, és máris jöhet az installálás.

Ha ez valami miatt nem sikerül, vagy nincs DOS a gépünkön, akkor kénytelenek vagyunk lemezről indítani. Ez történhet a csomagban kapott floppy lemezről, illetve a CD-n található bootlemezek floppyra írásával. Külön lemezek vannak a PCMCIA-s notebookkal rendelkezők részére a minél egyszerűbb installáció érdekében.

A sikeres bootolás után jelentkezünk be rootként, és indítsuk el a YaST-ot. Ha bárhol elakadunk, nyugodtan nyomhatunk [F1]-et, ekkor jól érthető magyarózó szöveget kapunk a képernyőre.

Miután eldöntöttük, hogy angolul vagy németül folytatjuk munkánkat, és választunk néhány apró kérdésre, particionáljuk a háttértárolónkat. Ha a particiók megvannak, akkor illesszük be ezeket a file-rendszerbe. Ha már vannak meglévő DOS, EXT2, VFAT particióink, akkor itt azokat is elhelyezhetjük. Ezt követi a számunkra fontos csomagok kiválasztása. Itt viszont nem ússzuk meg a floppy használatát, a kiválasztott listát lemezre kell írni, s a gépet újra kell indítani.

Újból indítsuk el a Linuxot, ahogy először tettük.

A YaST második kérdésére most mondjuk azt, hogy már van lemezünk elmentett adatokkal. Ennek alapján felinstallálja a kijelölt csomagokat a program. Ha ezzel megvagyunk, akkor egy kelleme menüből konfigurálhatjuk a LILO-t, hálózaton, időzónát és egyéb szabványkérdéseket. Ezek után készen állunk az első igazi indításra.

Ha első alkalommal rootként belépünk (mert más választásunk nincsen :-), akkor jön egy jelszóállítás, néhány újabb kérdés a YaST-tól az egérről és egyéb háziállatokról. Kilépéskor még kb. tíz percet vesz igénybe, amíg lefut néhány program: mandb, updatedb, fontlist az X-nek, TeX-nek stb. Ezalatt egy másik terminálon már lehet ismerkedni a rendszerrel, futtatni az Xf86configot, új kernelt fordítani (csak megszokásból, mert tökéletes az is, amit hozzá adnak :-).

## Beszerzési lehetőségek

Sokféle lehetőség áll rendelkezésünkre, ha be szeretnénk szerezni magunknak a disztribúciót. Egyrészt megvehetjük szépen dobozolja, CD-ROM-on, nyomtatott német vagy angol kézikönyvvel és 60 napos telefonos, e-mailes supporttal. Ennek listára 98 DM, egyetemi hallgatók számára 78 DM. A cikkírás időpontjában még nincs magyar viszonteladója a terméknek. A hagyományos utat kedvelők számára a cég címe:

S.u.S.E. GmbH  
Gebhardtstr. 2.  
D-90762 Fürth  
Germany  
Telefon: +49 911 7405331, fax: +49 911 7417755

Számítógépes postafiókjuk az Interneten: [suse@suse.de](mailto:suse@suse.de). Megfelelő kreditkártyával rendelkezők megrendelhetik a World Wide Weben keresztül is. Ehhez további információt lehet olvasni a cég németországi, illetve nemrég megnyílt amerikai honlapjáról: <http://www.suse.de>, illetve <http://www.suse.com> címen.

Akik kevés pénzzel, de jó Internet-kapcsolattal és jó nagy háttértárolóval rendelkeznek, azok ftp-n át is beszerezhetik maguknak a SuSE Linux-ot. Ehhez persze nem jár support, és hiányzik belőle egy-két olyan program, ami a dobozolt változatban megtalálható. Mindemellett nagyon jól használható. Letölthető Németországból az <ftp.gwdg.de/pub/linux/cdroms/suse>, illetve Magyarországon a [www.elte.hu/linux/SuSE](http://www.elte.hu/linux/SuSE) címeiről. ■

Szalai Károly  
([czw@rusalka.fornax.hu](mailto:czw@rusalka.fornax.hu))

# Debian Linux és telepítése

Sokakban felmerülhet a kérdés, hogy milyen az a Debian, ami napjainkban annyira felkapottá vált a Linux hívőinek körében. Ez a disztribúció is ugyanolyan szolgáltatásokat nyújt, mint a többi összeállítás, legfeljebb apróbb eltérésekkel.

A Debian 1993-ban hozta létre Ian Murdock, eleinte a Free Software Foundation's GNU project támogatásával. Manapság a Debian közvetlenül a GNU alá tartozik, és a „Software in the Public Interest” támogatásával készül. Természetesen minden Linux-alapú kernelen futtatható. Nevét Debra és Ian Murdock nevéből származtatják.

## A Debian Linux jellemzői

A Debian jelenleg több mint 700 csomagot tartalmaz. A telepítés során a felhasználó maga választhatja ki a telepítendő csomagokat a DSELECT segítségével. Minden egyes, a listában megtalálható csomagról részletes leírást kaphatunk.

Szabadon használható és átrendezhető: nem szükséges, hogy bármilyen vállalat, illetve társaság tagja legyen a felhasználó, fizetési kötelezettség nélkül dolgozhat vele. Minden csomag megváltoztatható, egészen addig, amíg az megfelel a GNU General Public Licence (GPL) leírásának. A Debianban kb. 50 csomag található (a mirror szervereken a non-free és a contrib könyvtárakban), amelyek megváltoztatása speciális feltételekkel lehetséges.

Dinamikus: körülbelül 200 önkéntes dolgozik folyamatosan a csomagok karbantartásán, és felügyelik az új kódokat. A Debian ennek megfelelően igen gyors ütemben fejlődik, általában három havonta új verziói jönnek a disztribúciónak, illetve az FTP-szervereket naponta frissítik.

## Különbség a Debian és a többi disztribúció között

Három lényeges különbség van a Debian és a többi Linux-disztribúció között:

A csomagok karbantartási rendszere (package maintenance system). Akár a teljes rendszer, akár egyes csomagok felújíthatók anélkül, hogy a meglevő formattálni kellene, vagy elvesztené a felhasználó a fontos konfigurációs file-jait, esetleg (egyes esetek kivételével) rebootolni kellene a számítógépet. A legtöbb Linux-disztribúció tartalmaz valamilyen csomagmenedzsert, de a Debiané egyedi és jól megtervezett.

Nyitott fejlesztésű. Amíg a többi Linux-disztribúció egyének, kicsi, zárt közösségek vagy kereskedelmi cégek által fejlődik, addig a Debian az első olyan Linux, amely az Interneten keresztül fejlesztést indította be, szem előtt tartva a szabad szoftverek és a Linux szellemét. A több mint 200 önkéntes 700 csomagnál is többet menedzsel és fejleszt. A Debian fejlesztői a legtöbb esetben nem új alkalmazások létrehozását, hanem a meglevők javítását és fejlesztését tűzték ki célul.

Mivel a Debian fejlesztői térben nagy távolságokra helyezkednek el egymástól, ezért szükségessé vált a hibák, illetve a hibajavítások gyors közreadása. Ezt leggyorsabban e-mailben, illetve a WWW-n keresztül lehet megtenni. Erről közelebbi információk a Bug-Log FAQ-jában találhatóak.

## Hardware-szükségletek

A Debian támogatja a teljes i386-os architektúrát, azaz az összes 32 bites PC-hez való Intel processzort, beleértve a 386, 486, Pentium és Pentium Pro és Pentium II processzorokat, valamint a vele kompatibilis AMD és Cyrix CPU-kat is.

Az 1.3.1-es Debian fejlesztői verziójában az m86-os processzorok (Motorola 680x0 processzorok, ahol x>2; MMU-val) már futnak, a Digital Alpha

## 1. LISTA: AZ FDISK LISTÁJA A SWAP PARTÍCIÓK LÉTREHOZÁSA UTÁN

Disk /dev/hdd: 4 heads, 38 sectors, 823 cylinders  
Units = cylinders of 152 \* 512 bytes

Device	Boot	Begin	Start	End	Blocks	Id	System
/dev/hdd1		1	1	10	741	82	Linux swap
/dev/hdd2	*	11	11	823	61788	83	Linux native

verzió kiadás előtt áll, a Sparc- és MIPS-változatok is a közeljövőben elkészülnek.

A telepítőkészlet feltételezi a 386-processzort, és támogatja a PCMCIA, illetve az SCSI kártyákat, amelyeket ismer a Linux. Hálózati támogatást beülthető modulokban nyújt, így nem szükséges azokat a kernelbe fordítani.

## RAM és lemezsükséglet

A Debian már 4 Mbyte RAM-mal is telepíthető. A legutóbbi installációs lemezek könnyen követhető útmutatást nyújtanak a kevés memóriával rendelkező gépek esetén is.

A profibbakk egy 386-os alaplapú gépen 4 Mbyte RAM és 40 Mbyte lemezerülettel már kielégítően futtathatják a Linuxot mind hálózatos, mind X11-szerverfunkcióit kihasználva. Ebben az esetben természetesen szükséges swap partíció is, egyébként a rendszer nem tud többfelhasználós rendszerként bootolni.

A rendszer *minimális* szükséglete (X-Window nélkül és csak 1-2 felhasználóhoz): 15 Mbyte terület az alaprendszerhez – ez egy minimális lehetőségekkel ellátott Unix, ami tartalmazza az alapvető hálózati támogatást is.

100 Mbyte szükséges a standard rendszerhez, illetve egyéb alkalmazások futtatásához a konzolon (X11 nélkül).

150 Mbyte szükséges a standard rendszerhez, és ahhoz, hogy X-Window alatti alkalmazásokat is használjunk.

300 Mbyte vagy annál nagyobb terület szükséges az összes opcionális program telepítéséhez.

Egy általános telepítéskor, feltételezve, hogy telepítünk néhány nagyobb programot, X-Window Systemet, illetve néhány felhasználót is szeretnénk kiszolgálni, a következő partíció nagyságokat javasolják:

- 20 Mbyte elég a rootnak, ha van külön /usr, /var, /tmp és /home partíció is;
- 500 Mbyte a /usr könyvtárnak (ha van hely, lehet közel 1 giga is, hogy jól elférjünk);
- 50 Mbyte a /tmp könyvtárnak, nagy szervereknél 100-ra is szükség lehet;
- 50–100 Mbyte a /var könyvtárnak, a levelezés mértékétől függően;
- a /home igény szerint, a felhasználók saját anyagai kerülnek ide;
- ~50 Mbyte swap partíciónak.

Az optimális swap disk megválasztása kritikus lehet a rendszer működésében. Általános megoldásként ajánlható, hogy a rendszer memóriájának kétszeresét válasszuk 32 Mbyte RAM-ig, afelett a memóriával megegyező legyen a swap. Természetesen ezek az értékek felhasználóink számától, illetve a futtatandó programoktól függően változhatnak.

## Hogyan partícionáljuk a merevlemezünket?

Természetesen ez is elvi kérdés, sok mindentől függhet. A felhasználók száma nagymértékben befolyásolhatja döntésünket. Az alábbiakban egy 1,6 Gbyte területű merevlemez partícionálási lehetőségeit láthatjuk:

- 30 Mbyte terület a root (/) könyvtárnak;
  - 450 Mbyte terület a /usr könyvtárnak;
  - 60-100 Mbyte swap területnek;
  - 900-1000 Mbyte a /home könyvtárnak;
  - 40 Mbyte a /var könyvtárnak;
  - a maradékot a /tmp könyvtárnak „adjuk”.
- Amennyiben nem akarunk külön partíciót a /tmp-nek, abban az esetben célszerű a /tmp könyvtárat egy szimbolikus linkkel a /var/tmp könyvtárhoz csatolni, ám ilyenkor annak ésszerűbb nagyobb területet is hagyni.

## 2. LISTA: AZ FDISK LISTÁJA A NATÍV LINUX PARTÍCIÓ LÉTREHOZÁSA UTÁN

Disk /dev/hdd: 4 heads, 38 sectors, 823 cylinders  
Units = cylinders of 152 \* 512 bytes

Device	Boot	Begin	Start	End	Blocks	Id	System
/dev/hdd1		1	1	10	741	82	Linux swap
/dev/hdd2	*	11	11	823	61788	83	Linux native

## A Debian és a nagy winchesterek

Van egy felső határ, hogy honnan lehet egy operációs rendszernek bootolni. Ez *nem* a Linux sajátossága, minden operációs rendszerben megtalálható ez a határ.

Alapesetben a BIOS nem tudja használni azokat a területeket, amelyek az 1024 feletti cylinder, illetve track felett vannak. Emiatt egyetlen operációs rendszer sem képes bootolni az 1 Gbyte feletti partíciókról. Emiatt szükséges, hogy az a terület, ahonnan a Linux bootol, e határ alatt legyen. Az összes többi terület ilyen szempontból lehet a határ felett is. Így egyetlen feltétel, hogy a /boot könyvtár (illetve célszerűen a / - gyökér - könyvtár, ami kis terület) az első 1024 sávban legyen. Egyébb információkért a Large-Disk mini-HOWTO-t érdemes elolvasni.

## PCMCIA-támogatás a Debianban

A PCMCIA kártyák támogatása David Hins fejlesztése. Ezek a támogatások nincsenek alpból beépítve a kernelbe. A pcmcia\_modules\* deb csomagok tartalmazzák azokat a modulokat és szolgáltatásokat, amelyek a kártyák használatához szükségesek. A csomag tartalmazza az azok telepítéséhez szükséges ismereteket is. Azokat a PCMCIA kártyákat, amelyek tartalmazznak IDE meghajtókat is, a kernel támogatja már a telepítés alatt is.

## Előkészületek a telepítéshez

A Debian-telepítő CD bootolható, azaz akinek megfelelő BIOS-a van a gépében ATAPI CD-vel, annak elegendő rábootolni a CD-re a telepítés megkezdéséhez. Ha nem jön össze a bootolás, akkor a CD \BO\DISKS-I3\MAGYAR\INSTALL.BAT indítása is

segíthet DOS - nem DOS-ablak - alól. Ha ez sem járható út, akkor a „floppyzgatás” segíthet az alábbiak szerint:

A Debian installálásakor meg kell nézni, hogy milyen meghajtó van abban a gépben, amire telepíteni szeretnénk. Ha 1,44-es meghajtóval rendelkezünk, akkor a következő file-okra van szükségünk: resc1440.bin, drv1440.bin.

Amennyiben a bootmeghajtónk 1,2-es, úgy a file-ok a következők: resc1200.bin, drv1200.bin. Mindkétfajta floppymeghajtóhoz kellenek még a BASE-1.BIN... BASE-5.BIN file-ok is.

Ezek a file-ok a \BO\DISK-I3\MAGYAR nevű könyvtáraiban találhatóak meg. (Ebben a könyvtárban angol nyelvű dokumentumok is megtalálhatók a telepítésről mind html, mind txt formátumban.) A fentebb felsorolt file-ok a telepítőlemezek image file-jai, és a teljes lemezt tartalmazzák raw formátumban. Egy speciális program segítségével írhatjuk vissza a tartalmukat a lemezre.

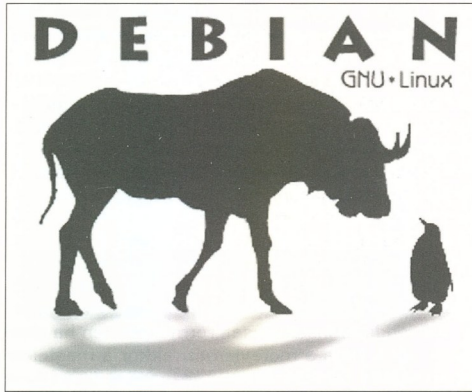
Ehhez szükségünk lesz egy, maximum hét formattált lemezre. A formattálás történhet DOS alatt is. A lemezek a következő neveket kapják:

- resc1440.bin vagy resc1200.bin → Rescue Floppy;

- drv1440.bin vagy drv1200.bin → A Device Drivers Floppy, csak akkor van rá szükségünk, ha régi CD-ROM-meghajtónkat nem ismerné fel elsőre a rendszer;

- base-1.bin → Base 1 és így tovább a Base 2... Base 5 floppyt nem kell külön írni, a telepítés során majd létre lehet hozni, ha szükség van rá. Általában CD-s telepítéshez nem szükségesek, csak hálózati telepítés esetén lesz szükségünk rájuk.

A /tools nevű könyvtárban található RAWRITE.EXE program segítségével írhatjuk át az image file-okat a lemezekre DOS, Windows vagy OS/2 alatt. A Rawrite használata nagyon egyszerű, elindítása után megkérdézi a file nevét és a célmeghajtót. Adjuk ki a „rewrite.exe file drive” parancsot,



ahol a file az image file neve, amit a drive (a:) meghajtóra szeretnénk írni.

A telepítőlemezek természetesen Unix, illetve Linux alatt is elkészíthetők. Mivel néhány rendszer automatikusan bemountolja a floppyt a rendszerbe, amikor a lemezt behelyezzük a meghajtóba, ezért kapcsoljuk ki ezt a módot, nincs rá szükségünk. Ezek után adjuk ki a „dd if=file of=/dev/fd0 bs=512 conv=sync ; sync” parancsot, ahol a /dev/fd0 annak a meghajtónak az elnevezése, amire írni akarunk, a file pedig az image file neve. Figyeljünk arra, hogy a lemezt addig ne vegyük ki, amíg a drive LED-je ég.

A telepítésnél az első nagy probléma a floppyk „megbízhatóságából” adódik. Amennyiben a telepítés során valamelyik lemezen hibát találunk, úgy egy másik lemezzel próbálkozunk, mivel az újrafarmázás gyakran nem segít.

Amennyiben a telepítőlemezek elkészültek, úgy helyezzük be a Rescue floppyt a meghajtóba, és reseteljük a gépet.

## Kezdünk neki!

Amennyiben CD-ről vagy floppyról sikerült elindítani a rendszert, akkor a képernyőn megjelenő „boot:” felirat az installálás kezdetét jelenti. Ezek után két dolgot tehetünk:

A funkcióbillentyűk segítségével különböző helpekhez juthatunk, illetve a prompt után a linux szót beírva paraméterezhető a bootolási folyamat.

Egy egyszerű [Enter] lenyomásával mindenféle

paraméter nélkül elindíthatjuk a Linux telepítését. Ezek után egy „Loading...”, majd „Uncompressing Linux...” feliratnak kell megjelenni, amit számtalan sok üzenet követ. Általában elég sok hibaüzenet keveredik ezek közé, de most hagyjuk őket figyelmen kívül: gyakorta azt jelentik, hogy a telepítőlemezben található kernel olyan device-okat keres, amelyek nincsenek a rendszerünkben. Ezt a későbbiekben javíthatjuk egy saját beállításainknak megfelelő kernellel.

Amennyiben a gépben kevés memória van, abban az esetben egy szakaszt látunk egy szöveges menüben, három választással. Ekkor más módon történik az installálás, mert még a telepítőmenü elindulása előtt particionálni kell a winchestert, létre kell hozni egy swap partíciót, amit aktivizálni is kell, majd ezután indíthatjuk a menüs installálási procedúrát. Ezt a swap partíciót a Linux a kellő mennyiségű memória pótlására fogja használni. A swap partíció létrehozásához a *cfdisk* nevű program használható, melyről bővebb információt a *cdisk.txt* kézikönyvben találhatunk.

Amennyiben a bootolás sikeres volt, úgy egy párbeszédpanelt láthatunk a képernyőn, melyen kijelölhető, hogy színes vagy fekete-fehér telepítést kívánunk. A választás után a főmenübe érkezünk.

Ez az a menü, amivel az érdemi installálás során a legtöbbet fogunk találkozni. A rendszer felméri, hogy jelenleg hol állunk a telepítéssel, és ennek megfelelően kapunk egy menüt. A menü legfelső választási lehetősége a rendszer felmérése utáni következő pont, amit érdemes elvégezniük. Gyakorlatilag a telepítés során ezen menüponton [Enter]-eket ütve, szinte minden erőfeszítés nélkül juthatunk át az akadályokon.

Ha eddig eljutottunk, akkor a billentyűzet konfigurálása következik. Válasszuk ki a kívánt billentyűzet típusát, illetve a hozzá legközelebb esőt. Későbbiekben igen széles skálán lehet majd kiválasztani a nekünk megfelelőt, amennyiben most nem találtuk volna meg.

Természetesen a Linux lehetővé teszi, hogy a telepítést több szempontból is figyelemmel kísérjük. Amennyiben a bal oldali [Alt] billentyű lenyomva tartása mellett valamelyik funkcióbillentyűt leütjük, akkor különböző virtuális terminálokon követhetjük nyomon az eseményeket:

Az F2-es terminálon például a Linux shelljét kapjuk, ami ugyan egy lebutított Bourne-shell klón, de azért alapjaiban igen jól használható. Megnézhetjük a használható parancsok listáját, az *ls /bin /sbin /usr/bin /usr/sbin* paranccsal.

Általában jól működő install lemezek esetén ennek a shellnek a használata nem szükséges, inkább a menüt használjuk. Visszaváltani a bal [Alt] és az [F1] lenyomásával lehetséges. A Linux egyébként maximum 64 virtuális konzolt támogat, a telepített alaprendszeren 6 ilyen van, ami természetesen bővíthető.

## Particionálás

Elérkeztünk a telepítés legbonyolultabb részéhez, a merevlemez particionálásához. Ha olyan adatok vannak a merevlemezen, amelyek fontosak, akkor lépünk ki a telepítésből, és mentsük le őket. A particionálásához a már említett *cfdisk* programot használjuk. A *cfdisk* a RedHat és a Slackware „kőbalta” fdiskjével szemben menüs program, így használata nem okozhat gondot annak, aki már valaha particionált winchestert.

A Linux minimum két partíciót igényel alapesetben. Az első feladatunk, hogy létrehozunk a swap partíciókat. Ez gyakorlatilag egy virtuális memóriát jelent, melynek nagysága általában 16 és 128 Mbyte között legyen.

Ellentétben egyes elterjedt rendszerekkel, a Linux több elsődleges partíciót képes kezelni, szám szerint négyet. Ezek bármelyike helyett készíthetünk egy-egy kiterjesztett (extended) partíciót is, mely több logikait is tartalmazhat, így apró részekre szabdalhatjuk a winchestert. Természetesen ha csak mi magunk szeretnénk használni a Linuxunkat, akkor elég egy swap és egy Linux partíció.

Először is hozzuk létre a swap partíciókat, ami a normál memóriánk kétszerese legyen, illetve 32 Mbyte felett, vele megegyező (ettől el lehet térni, ez csak ízlés, illetve használat kérdése). Amennyiben csak két partíciót hozunk létre, úgy mindkettő lehet primary. Több partíció esetén ügyeljünk, hogy a 3. után már csak extended lehet, így az utolsót célszerű nagyobbra hagyni.

A swap partíció elkészítése után hozzunk létre egy natív partíciót is, amire a tulajdonképpeni Linux kerül. A merevlemez ekkor a 2. lista szerint nézhet ki, természetesen más-más számokkal, a mérettől függően.

Tegyük bootolhatóvá a Linux natív partíciókat, majd lépünk ki.

Ezzel visszatérünk a főmenübe, ahol a következő lépés a swap aktiválása. A rendszer detektálja, hogy melyik partíció lett kiválasztva, majd elfogadhatja velünk. Választási lehetőségként megvizsgálhatjuk,

hogy minden szektorunk megfelelő-e, de erre csak MFM, RLL vagy régi SCSI merevlemez esetén van szükség, IDE típusú esetén soha. *(Az ellenőrzés soha nem árt, láttunk már új 2 Gbyte-os HDD-t is hibás szektorokkal tele! – a szerk.)* Az aktiválás után a partíciók virtuális memóriaként üzemel a rendszerünkben, ezért kell ezt inicializálni először.

A főmenü következő lépése a Linux natív partíció aktiválása, ami hasonló az előzőekhez. A merevlemez vizsgálata itt sem árt, de lassú rendszereknél több tíz percig is eltarthat. Az első partíciót a rendszer a gyökérfájlistának fogja mountolni, ami a / jellel lesz elítva. A winchester formattálása után már ext2 filesystem van a lemezen, és következhet az alaprendszer installálása. Természetesen ha több részre lett bontva a merevlemez, akkor a megfelelő menüpontok kiválasztása után azokat is fel kell mountolni.

Ha megtörtént a mountolás, akkor már csak az alaprendszer felrakása van hátra, illetve néhány beállítás. Válasszuk a következő menüpontot, és szépen várjuk ki, amíg a CD-ről „felsétál” a rendszer. (Ha nincs CD, vagy a telepítő által nem támogatott eszközről akarunk telepíteni – pl. hálózati telepítés esetén –, szükség lehet a Base 1-4 floppykra is.)

## A kernel beállításai

A következő lépés a kernel installálása. „Floppy-zás” esetén, ha nem az lenne benne, akkor helyezzük vissza a Rescue floppyt a meghajtóba, majd a rendszer felmásolja a kernelt a merevlemezre. A későbbiekben majd lehetőségünk lesz arra, hogy egy saját bootdisket hozzunk létre, és bootolhatóvá tegyük a gépünket floppy nélkül is.

Amennyiben van speciális kártyánk, pl. PCMCIA, úgy most a device driverek installálása következik, ellenkező esetben ezt kihagyhatjuk. Ennél a menüpontnál kiválaszthatjuk, milyen speciális driverek töltdőjenek be a boot folyamán.

Ezek után történik a rendszer lényeges jellemzőinek beállítása. Elsőként a rendszerórát állítuk be, célszerűen a Europe/Budapest kijelölésével. A zónaidő kiválasztásakor figyelni kell arra, hogy találató-e más operációs rendszer is a gépen. Amennyiben csak a Linux van, úgy a GMT opcióit válasszuk. Ebben az esetben a rendszer órája lesz a GMT ideje, és a zónaidőt a Linux programból generálja. Amennyiben más rendszerek (mint pl. DOS vagy Windows) is találhatóak a gépen, úgy a localtime-et válasszuk.



## 3. LISTA: EGY PÉLDA LILO.CONF FILE

```

boot = /dev/hda           # a primary masterről bootolunk
delay = 20                # két másodpercig vár
read-only                 # csak olvasható
    image = /boot/vmlinuz # a default kernel a vmlinuz
    label = Linux          # a Linux opció
    root = /dev/hda3      # vinchester 3. partíciója a gyökér
    vga = normal          # normál video mód
other = /dev/hda1         # a másik rendszer, most éppen DOS
    label = dos            # Dos a címkéje
    table = /dev/hda      # a bootoláshoz használatos infókat
                          # master boot rekordból veszi

```

A következő lépésben a hálózat konfigurálása történhet meg. Amennyiben nincs hálózatra kötve a gépünk, úgy nyilvánvalóan csak a gép nevét firtató kérdésre kell válaszolni, majd az „Is your system connected to a network?” kérdésre nemet. *(Ez azóta magyarra lett fordítva a CD-n! – a szerk.)*

Hálózatos felhasználás esetén azonban több kérdés is felmerül, amelyek megválaszolásához célszerű a hálózat rendszergazdjától több információt beszerezni. Ezek:

- a számítógép neve (saját magunk adhatjuk meg);
- a domain név, ezt már a rendszergazda adja meg;
- a gépünk IP-címe, ami szintén korlátozások alá eshet;
- a hálózatban használt netmask;
- a hálózat IP-címe;
- amennyiben a hálózatban van gateway, akkor annak az IP-címe;
- a DNS-szerverként használatos gép címe;
- szabad-e csatlakozni a hálózathoz.

Ezek közül az információk közül nem mindre van szükség. A későbbi beállítás esetében a /etc/init.d/network nevű scriptet kell módosítani.

Ezek után történhet a bootolás beállítása. Célszerűen az installálásnál a Linux bootmanagerét (LILO) a master boot rekordba kell installálni. A LILO segítségével kevésbé látványosan, de jól használhatóan lehet több operációs rendszert is bootoltatni. A későbbiekben látható lesz egy többrendszeres beállítás leírása. Bővebb információkat a /usr/doc/lilo alatt lehet találni. Ezek után az utolsó lépés a boot floppy elkészítése, majd a rendszer indítása.

Fontos! A Linuxot SOHA nem szabad kikapcsol-

ni, amíg a rendszer leállítása meg nem történt! Ez a [Ctrl Alt Del] kombinációval történhet, illetve a *shutdown* parancs segítségével. Tehát az újraindítás után meglátjuk, hogy mit csináltunk...

## A rendszer végleges telepítése

A bootolás után legelőször a root, vagyis a super user jelszavának beállítása következik. Minden jelszóra igaz, hogy minimum 6, maximum 8 karakter lehet alapesetben, és felváltva kell tartalmaznia kis- és nagybetűt, illetve számokat.

Másik fontos figyelmeztetés: csak akkor lépünk be rootként, ha a rendszert karbantartjuk, illetve adminisztráljuk. Minden esetben a saját azonosítónkat használjuk, amit a rendszer azonnal meg is csináltat velünk. Erre azért van szükség, mert így gyakorlatilag kizárható a vírusfertőzés, illetve a külső károkozás esélye. Emiatt a Linux gyakorlatilag vírusmentes rendszer egyéb operációs rendszerekkel szemben, amelyek minden programjukhoz root-jogot adnak.

Ezek után van egy működőképes rendszerünk, amely még csak az alapszolgáltatásokat ismeri. A többi csomag telepítéséhez célszerű a *dselect* nevű program használata, ami egy igen intelligens csomagkezelő rendszer. Elindítása után az első lépésként kiválaszthatjuk, honnan szeretnénk a csomagokat telepíteni. Hat választásunk van, de lehetőleg a CD-ROM-ot válasszuk. Ezek után rákérdez a CD-ROM-unk elérési útvonalára, és itt megint egy kis kitérő szükséges:

A Linux mindent file-ként kezel. File minden meghajtó, port és eszköz. A merevlemezeknél az IDE vezérlők esetén a primary master merevlemez

a /dev/hda, a slave a /dev/hdb és így tovább a /dev/hdc és /dev/hdd meghajtókig.

Tehát ha a CD-nk pl. a secondary master meghajtó, akkor ide /dev/hdc-t kell írunk. Az SCSI vezérlőknél /dev/sda a helyes megadás, a CD-ROM pedig a /dev/scd1-re hallgat. Az egyes partíciókra a számmal hivatkozhatunk. A secondary master meghajtó második partíciója így a /dev/hdc2 nevet kapja meg. A CD-ROM-ot nem(!) kell külön felmountolni az installáláshoz, ez ekkor automatikusan történik meg. Ezek után a program rákérdez a megfelelő könyvtárak neveire, amelyeket előzőleg célszerű leírni magunknak. Ha minden kérdésre jól választunk, akkor a következő lépésben a rendszer automatikusan frissíti a saját adatbázisát, amire az upgrade során lehet szükség.

Harmadik lépésként a kiválasztás történik meg, hogy mely csomagokra van szükségünk a használatnál. A képernyő két részre oszlik, alul az épp aktuális csomag leírása található. A [d] billentyű lenyomására az információk egyre bővülnek, így könnyítve meg a választást. Egyes csomagok telepítésénél előfordulhat, hogy más csomagok is szükségesek, így ezek kiválasztásához egy újabb, az eddigiehez hasonló menübe jutunk.

A csomagok kiválasztása után [Q]-val térhetünk vissza az eredetiekhez, [X]-szel pedig az kiválasztások elhagyásával megyünk a kiindulási ponthoz. Fontos a kis- és nagybetűk megkülönböztetése!

Javaslatként érdemes először a legtöbb csomagot felrakni, majd később, ha már átlátjuk a rendszert, a feleslegeseket letörölni. A csomagok kiválasztásánál a [-], [+] és [\_] billentyűk használatok.

Ezek után történik a csomagok installálása, ami egy viszonylag hosszabb művelet lesz. Az installálás után a csomagok konfigurálása következik, amit sajnos nem áll módomban részletezni, egyrészt a nagy számosság, másrészt a variációk végtelen száma miatt.

Mivel a Linux az egyik legjobban dokumentált rendszerek egyike, ezért az online kézikönyvek sokat segíthetnek a megértésében. Természetesen itt már használható több virtuális konzol, a fent leírt módszer szerint (bal [Alt] és az [F1]-[F6] funkcióbillentyűkkel való váltás). Alapesetben hat konzolunk van, de ezek száma bővíthető.

## A LILO felrakása

Végül ígéretemhez híven következzenek egy LILO (Linux LOader) beállítás a több operációs rendszerrel rendelkező gépekre.

Először is valamilyen szövegszerkesztő segítségével editálni kell a /etc/lilo.conf elnevezésű file-t. Mazochista típusoknak a vi nevű ajánlott, ami végtelen sok funkcióval rendelkezik, de ennek megfelelően kellően nehézkes a használata (egyések szerint olyan, mint egy krómozott nyelv kóbalta). Akik szeretik a kezelhetőbb, egyszerűbb szerkesztőket, azok számára a Turbo Pascal szerkesztőjéhez hasonló joe-t javaslom. Természetesen ezeket a programokat is telepíteni kell előzőleg!

Tehát feltételezve, hogy kevésbé önsanyargatóak vagyunk, adjuk ki a „joe /etc/lilo.conf” parancsot. (Természetesen ezt csak rootként lehet editálni!) A file-t a 3. lista alapján szerkesszük meg.

Amennyiben megtörtént a szerkesztés, adjuk ki a *lilo* parancsot, ami helyes *lilo.conf* esetén a következő üzenetet válaszolja: „Added Linux \*” majd „Added dos”. Tehát az elsődleges bootolás a Linux címkéről történik. Ezek után ha újraindítjuk a rendszert (CTRL-ALT-DEL!), akkor a LILO: promptnál egy Tab billentyűt nyomva megkapjuk a választási lehetőségeket, jelen esetben a „Linux” és „dos” feliratokat. Két másodperc várakozás után (delay = 20) automatikusan a Linux fog indulni.

Ennyit előljáróban a Debian installálásáról, és jó szórakozást mindenkinek! ■

Gombos Márton  
([nihil@makos-teszt.sote.hu](mailto:nihil@makos-teszt.sote.hu))

# Programozzunk X-Window alatt!

Sokan kérdezik, hogy mennyire bonyolult az X-Window alá programokat írni. A most következő cikkben elkészítjük a közismert amőba játék linuxos verzióját X-Windowhoz, hátha ettől másnak is kedve támadt ezt a „bonyolult” rendszert programozni.

## A widget-készletek világa

A Linux hiába a világ egyik legizgalmasabb operációs rendszere az otthoni PC-kre, hiába tud olyan paramétereket felmutatni, melyeket messziről megcsodálnak és sárga irigységgel agyonhallgatnak más operációs rendszerek tulajdonosai, ha a kezelői felülete a 70-es években használt karakteres konzolra szűkülne, nem lenne sok esélye arra, hogy széles tömegeket meghódítson. De szerencsére a Linuxhoz szabadon használhatóan megkaphatjuk az X-Window rendszert, mely egy igen hatékony grafikus felületet ad a Linuxot használók kezébe. A Linuxot használók többsége nap mint nap használja ezt a grafikus felületet, és páran közülük néha elgondolkodnak azon, hogy milyen jó lenne olyan programokat írni, amelyek az X-Window felületen futnak.

A legtöbb X-program C-ben íródott az X-Window rendszer alapfunkció-gyűjteményét felhasználva, melyben az alapvető rendszerhívások találhatóak. Ezekkel a rendszerhívásokkal, melyek az Xlib library file-ban találhatóak, egy X-program egy X-felületen létrehozhat különböző ablakokat, kezelőszerveket és más grafikus elemeket. De ez a módszer nagyon sok programozási munkát igényel, mert az Xlib-ben nagyon alacsony szintű rutinok találhatóak. Pont azért, hogy ne kelljen sok munkát befektetni egy X-program létrehozásába, jöttek létre az úgynevezett widget-készletek, melyek egyszerű a programozó dolgát könnyítik meg, másrészt pedig mindazok a programok, melyek egy közös widget-készletet használnak, hasonló kinézettel és ke-

zeléssel büszkélkedhetnek. Sok ilyen widget-készlettel találkozhatunk az X-rendszerhez. Talán az egyik legerjedtebb a Motif, mely sajnos csak pénzért szerezhető be. Ezenkívül még ha körbenézünk linuxos berkekben, ráakadhatunk a Qt widget készletre is, mely a C++ programozási nyelvből teszi az X-felületet könnyebben használhatóvá. Én most ennek a cikknek a keretein belül az xforms widget-készletből fogok egy kis bemutatót tartani, melynek segítségével C-ben írhatunk rövid idő alatt igazán tetszetős programokat.

## Az xforms igazán megkönnyíti a munkát

Az xforms widget-készlethez egyetlen egy include file tartozik, mégpedig a forms.h. Ebben az include file-ban vannak felsorolva mindazok az elemek, melyeket a xforms megalkotói számunkra előre elkészítettek. Ilyen elemek pl.:

- maga az alapablak (Forms /vagy Windows),
- különböző négyszögletes területek (Boxes),
- menük (Menus & Choices),
- listázók (Browsers),
- szövegek (Texts),
- nyomógombok (Buttons),
- húzógombok vagy potméterek (Sliders),
- bekérési mezők (Input fields),
- grafikák (Bitmap & Pixmap),

és még sok minden más.

Egy xforms-t felhasználó program tehát legelőször is be kell hogy includolja a forms.h-t, a szokásos

```
#include <forms.h>
```

módon.

De még mielőtt továbbmennénk, pár szót ejtenék arról, hogy a következőkben leírt programok mind megtalálhatók a példaprogramok között. A most tárgyaltakat kövessük nyomon az xamoba\_01.c programban. A cikkben leírt procedúrák pontos szintaxisát az olvasó az xforms-hoz adott leírásokból és a forms.h-ből maga is kikeresheti, ezért ezt a cikkben a program megérté-

séhez szükséges mértéken túl nem magyarázom. Magában a programban az első teendő, hogy létrehozunk a kapcsolatot az X-szerverrel. Ezt az `fl_initialize` procedúrával tehetjük meg.

```
int main(int ac, char *av[])
```

```
{
    Display *dpy;
    dpy = fl_initialize( &ac, av, MY_PRGNAME, 0, 0 );
```

Miután felépült a kapcsolat, elkezdhetjük a tényleges munkát. Először is hozzunk létre egy ablakot:

```
#define MY_WIN_X 400
#define MY_WIN_Y 400
FL_FORM *my_form;
my_form = fl_bgn_form( FL_UP_BOX, MY_WIN_X, MY_WIN_Y );
```

Ezzel a paranccsal létrehoztunk egy szürke hátterű, teljesen kitöltött ablakot, ami még nem jelent meg a képernyőn. Ahhoz, hogy megjelenjen a képernyőn, két dolgot kell még tennünk. Először is be kell fejeznünk a form és a hozzá tartozó objektumok definiálását:

```
fl_end_form();
```

másodszor pedig meg kell jelenítenünk:

```
#define MY_PRGNAME "xamoba"
fl_show_form( my_form, FL_PLACE_SIZE, FL_FULLBORDER,
MY_PRGNAME );
```

Az itt megadott paraméterek döntik el, hogy végül is a képernyőn hol és milyen módon jelenjen meg az ablak. A fent használt paraméterekkel egy kerettel körbevett ablakot kapunk, melyet teljes egészében az általunk használt window-manager fog elhelyezni a képernyőn. Az ablak tetején az „xamoba” feliratot lehet majd olvasni. Legalábbis lehetne, ha nem fejeződne be a program egy szempillantás alatt. Ahhoz, hogy ne egy felvillanással tűnjön el a képernyőről, a programot várakozásba kell vinnünk. Mint később látni fogjuk, egy X-es programban általában egy a felhasználó által kezdeményezett eseményre várunk és reagálunk. Ezeket az eseményeket, melyeket az egérmutató mozgása, az egér valamelyik gombjára való kattintás, vagy esetleg egy billentyű megnyomása a billentyűzetten okoznak, idegen kifejezéssel eventnek (esemény) hívjuk. Az xforms lehetőséget ad nekünk arra, hogy ilyen eventekre várakozzunk, és arra is, hogy csak bizonyos előre definiált eseményekre várjunk. Legegyszerűbb módja a várakozásnak az

```
obj = fl_do_forms();
```

procedúra, amely csak akkor adja vissza a vezérlést a programunknak, ha egy általunk létrehozott objektumot a felhasználó valamilyen módon triggerrel. Ilyen pl. egy kattintás egy nyomógombra, vagy egy potméter elhúzása valamelyik irányba. Minden más esetben a program várakozásban marad.

Legvégül pedig, a program befejezése előtt, az általunk létrehozott objektumokat el kell tüntetni a képernyőről:

```
fl_hide_form( my_form );
```

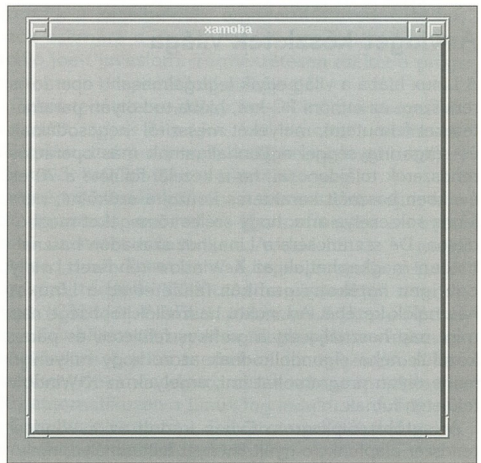
és az általuk lefoglalt memóriaterületeket is fel kell szabadítanunk:

```
fl_free_form( my_form );
```

```
return 0;
```

```
}
```

Ezzel meg is írtuk az első programunkat (xamoba\_01.c), melyet ha lefuttatunk, akkor egy szürke hátterű ablakot kapunk, melyen semmiféle kezelőszerv nincsen (1. kép). Magát a programot is csak az ablak bal felső sarkába kattintva a window-manager által felkínált `close` vagy `quit` opciókkal lehet befejezni, mert az `fl_do_forms()` procedúra soha nem tér vissza a várakozásból, hisz egyetlen objektumot sem hoztunk létre az ablakban.



## Díszítés és kapcsolók

Egy szürke hátterű program bár jól használható, de azért szebb lenne, ha kiszínezhetnénk. Ehhez az ablakunkban kijelölünk néhány négyszög alakú területet és beszínezzük őket, illetve szöveget írunk beléjük. Az első ilyen terület, mely a játéktér háttéréül fog szolgálni:

```
obj = fl_add_box( FL_BORDER_BOX, 0, 50, MY_WIN_X-100,
MY_WIN_Y-50, "" );
```

Miután létrehoztuk a megfelelő koordinátákon a megfelelő mérettel, színezzük is ki:

```
fl_set_object_color( obj, FL_SLATEBLUE, FL_COL1 );
```

Szép sötétlila négyszöget kapunk végeredményül. Magát az objektumot nem jegyezzük meg, mert a továbbiakban nem fogjuk megváltoztatni, és eventekre sincsen szükségünk tőle. Ezt a négyszögletes részt kizárólag csak díszítésként használjuk. Ezért a létrehozáskor az általános obj változóba jegyezzük meg az objektumra mutató pontintert, és eme mutató segítségével közvetlen a létrehozás után állítsuk be az objektum paramétereit, amelyeket az objektum a program egész élete folyamán megtart majd. Mivel a pontintert később az obj változóban felülírjuk, így későbbi változtatásra nem lesz majd lehetőség.

Most hozzunk létre egy fejléctet, melybe írjuk bele a program nevét:

```
#define STR_FEJLEC "X-Amóba v0.01"
obj = fl_add_box( FL_BORDER_BOX, 0, 0, MY_WIN_X, 50,
STR_FEJLEC );
```

Színezzük ki a négyszögletes tartományt:

```
fl_set_object_color(obj,FL_DARKGOLD,FL_COL1);
```

Állítsuk be a szöveg méretét és stílusát:

```
fl_set_object_lsize( obj, FL_HUGE_SIZE );
```

```
fl_set_object_lstyle( obj, FL_BOLD_STYLE );
```

Végezetül pedig állítsuk be a szöveg színét is:

```
fl_set_object_lcol(obj, FL_BLACK);
```

Most, hogy készen vagyunk a fejléccel és a játéktérrel, hozzunk létre még egy négyszögletes területet a kezelőszervek részére:

```
obj = fl_add_box(FL_BORDER_BOX, MY_WIN_X-100, 50,
100, MY_WIN_Y-50, "" );
```

```
fl_set_object_color(obj,FL_DARKCYAN,FL_COL1);
```

A következő lépés a kezelőszervek létrehozása. Alakítsunk ki három nyomógombot. Mégpedig úgy mint: [Új játék] [Ismertető] és [Vége]. A [Vége] nyomógomb kialakítása:

```
#define STR_VEGE "Vege"
FL_OBJECT *vege_gomb;
vege_gomb=fl_add_button( FL_NORMAL_BUTTON,
MY_WIN_X-90, 50+120, 80, 30,
STR_VEGE );
```

Evvél létrehoztuk a nyomógombot, amely szürke színű. De a szöveg, amit rajta olvashatunk, eléggé kicsi betűkkel jelenik meg. Ezért válasszunk egy kicsit nagyobb betűkészletet hozzá:

```
fl_set_object_lsize( vege_gomb, FL_NORMAL_SIZE );
```

A másik két nyomógombot is hasonló módon hozzuk létre:

```
#define STR_UJ_JATEK "Új játék"
FL_OBJECT *uj_gomb;
uj_gomb=fl_add_button( FL_NORMAL_BUTTON,
```

```
MY_WIN_X-90, 50+20, 80, 30,
STR_UJ_JATEK );
fl_set_object_lsize( uj_gomb, FL_NORMAL_SIZE );
```

```
#define STR_INFO "Ismertető"
FL_OBJECT *info_gomb;
info_gomb=fl_add_button( FL_NORMAL_BUTTON,
MY_WIN_X-90, 50+70, 80, 30,
STR_INFO );
```

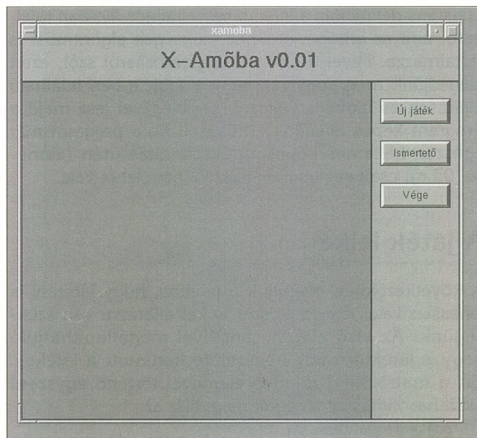
```
fl_set_object_lsize( info_gomb, FL_NORMAL_SIZE );
```

Mint ahogy észrevehettük, itt már megjegyezzük a nyomógombokra mutató pontintereket. Erre azért van szükség, mert a későbbiekben ha valaki megnyomja a gombok egyikét (rákattint az egérrel), az `fl_do_forms()` eljárás által visszaadott mutatóval összehasonlítva a le-tárolt mutatókat megtudhatjuk, hogy melyik nyomógombot nyomta le a felhasználó. Ez pl. történhet így:

```
do
{
obj = fl_do_forms();
if ( obj == vege_gomb ) break;
} while(1);
```

Vagyis mindaddig, amíg nem a [Vége] gombot nyomta meg a felhasználó, egy végtelen ciklusban vagyunk, melyben a program az `fl_do_forms()` eljárásnál várakozik az eventekre.

A programot (`xamoba_02.c`) lefordítva és futtatva most már egy színes ablakot kapunk, melynek fejléce van és három nyomógombja a jobb oldalon (2. kép). A [Vége] nyomógombra kattintva a program definiáltan befejeződik.



## A nyomógombokról egy picivel többet

Az amőba játék egy 19×19-es négyzethálós pályán folyik. Ehhez létre kell hoznunk egy ilyen pályát. Legegyyszerűbb módja, ha nyomógombokból alakítjuk ki, mely nyomógomboknak a tulajdonságát a célnak megfelelőre változtatjuk. Vagyis vastag fekete keret és fehér háttér, viszonylag nagyméretű betűkkel a két játékost jelképező X és O karaktereknek. A nyomógombokra mutató pointereket egy kétdimenziós tömbbe rendezve tároljuk, hisz a nyomógombok eventjeire is szükségünk lesz majd a későbbiekben.

```
#define NEGYZET 20
FL_OBJECT *matrix[N][N];

for ( x=0; x<N; x++ )
{ for ( y=0; y<N; y++ )
  { matrix[x][y]=fl_add_button( FL_NORMAL_BUTTON,
(NEGYZET/2)+(x*NEGYZET),
50+(NEGYZET/2)+(y*NEGYZET),
NEGYZET, NEGYZET,
"" );
  fl_set_object_boxtype( matrix[x][y],
FL_BORDER_BOX );
  fl_set_object_color( matrix[x][y],
FL_LEFT_BCOL, FL_MCOL);
  fl_set_object_lstyle( matrix[x][y],
FL_FIXEDBOLD_STYLE);
  fl_set_object_lsize( matrix[x][y],
FL_LARGE_SIZE );
}
}
```

A kétdimenziós tömb definíciójához felhasznált N konstans definíciója a logic.h nevű include file-ban található, ami az amőba játékhoz szükséges algoritmust is tartalmazza. Mivel a cikk X-programozásról szól, ezért elégedjünk meg annyival, hogy a logic.c-ben található eljárások és globális változók segítségével lesz majd a program képes ellenünk amőbázni. Ez a program már egészen kellemes képet mutat elindítás után (xamoba\_03.c), csak sajnos még játszani nem lehet vele.

## A játék lelke

A következőkben beépítjük a játékba, hogy játszani is lehessen vele. Ehhez először is két eljárásra van szükségünk. Az első eljárás, amellyel megállapíthatjuk, hogy a játéktér mely négyzetére kattintott a játékos. Ezt a matrix[x][y] minden elemével történő egyszerű összehasonlítással tehetjük meg, hisz az

```
obj = fl_do_forms();
```

arra az objektumra ad vissza egy mutatót, amelyikre kattintottunk. Ezt az összehasonlítást tegyük egy procedúrába, és a matrix[x][y]-t tegyük globálisná, hogy a procedúrából is elérhető legyen.

```
int GetKoordinat( FL_OBJECT *obj, int *x, int *y )
{ int a,b,ret;
  *x=-1;
  *y=-1;
  ret=-1;
  for ( a=0; a<N; a++ ) {
    for ( b=0; b<N; b++ ) {
      if ( matrix[a][b] == *obj ) { *x=a; *y=b;
ret=0; };
    };
  };
  return(ret);
}
```

Vagyis, ha mégsem egy olyan objektumra kattintottunk ami a játéktér részét képezi, akkor az eljárás -1-gyel tér vissza.

A másik eljárás, amit meg kell írunk, egy játékost jelenít meg a játéktér egy mezőjében. Ehhez előbb el kell döntenünk, hogy pontosan mit is jelenítsen meg, és hogy hogyan. Ehhez a következők kis táblázatot rajzoljuk:

–	előtér	–	háttér	–
–	karakter	szín	normál	kiemelt
egyik (kezdő) játékos	X	kék	fehér	szürke
másik (védekező) játékos	O	piros	fehér	szürke

Ennek alapján a következő procedúrát készíthetjük el:

```
#define HIGH 0
#define NORMAL 1
void p_Negyzetetbe( int x, int y, int z )
{
  /*== Ha nincs mit kirakni akkor vége és gyorsan vissza ==*/
  if ( (x==-1) || (y==-1) ) return;

  /*== Háttérszín ==*/
  if ( z==HIGH )
    fl_set_object_color( matrix[x][y], FL_MCOL,
FL_MCOL);
  else
    fl_set_object_color( matrix[x][y], FL_LEFT_BCOL,
FL_MCOL);
}
```

```

/*== Előterszín es forma ==*/
if ( Board[x][y]==human )
{
    fl_set_object_label( matrix[x][y], "X");
    fl_set_object_lcol( matrix[x][y], FL_BLUE );
}
else if ( Board[x][y]==computer )
{
    fl_set_object_label( matrix[x][y], "O");
    fl_set_object_lcol( matrix[x][y], FL_RED );
}
else
{
    fl_set_object_label( matrix[x][y], " ");
    fl_set_object_lcol( matrix[x][y], FL_BLACK );
}
}

```

Ahol a bemenő paraméterek alapján (x,y koordinátái a mezőnek, és a megjelenítés módja HIGH vagy NORMAL) a matrix[][] tömbben tárolt pointerok segítségével a megfelelő objektum által megjelenített karaktert, a karakter előtér- és háttérszínét megváltoztatjuk.

Ezek után nekiülhetünk a játék lefolyását vezérlő rész megírásának, melynek vázlatos folyamatábráját a 3. képen láthatjuk. Mint a folyamatábrából láthatjuk, van még egy rész, amiről eddig még nem ejtettem szót. Ez pedig a játék végén található üzenetküldés.

Ez többféle módon elképzelhető, mi most válasszuk a legegyszerűbbet, amit az xforms létrehozói számunkra felkínálnak. Küldjük a számítógép előtt ülő játékosnak egy dialógablakot, amibe írjuk be a megfelelő szöveget:

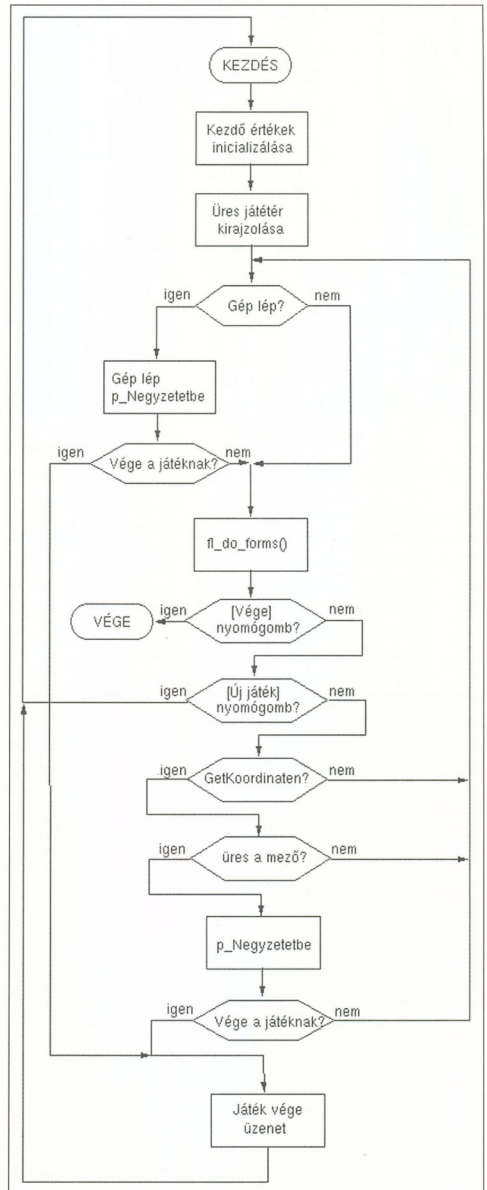
```

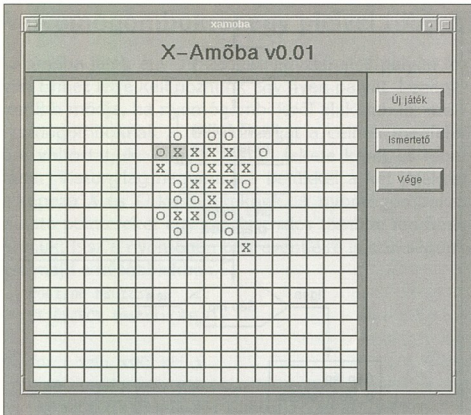
#define STR_HUMAN_WIN    "Gratulálok! Nyertél!"
#define STR_COMPUTER_WIN "Hmmm... hát gyakorolj még!"

if ( fGameOver )
{
    if (actual==player1)
    {
        fl_show_message( STR_FEJLEC, "", STR_HUMAN_WIN );
    }
    else
    {
        fl_show_message( STR_FEJLEC, "",
STR_COMPUTER_WIN );
    }
}

```

Ha most a játékot elindítjuk (xamoba\_04.c), akkor már játszhatunk a gép ellen.





## Építsünk dialógusablakot

Eddig a játék felületén lévő [Ismertető] nyomógombhoz semmiféle funkciót nem rendeltünk. Most készítsünk hozzá egy dialógusablakot, melyben a játékot használó információkat kaphat a játékról. Ehhez készítsünk egy teljesen önálló új formot, mely egy nyomógombot (amivel vissza lehet térni a játékhoz), egy listázó menüt (melyen a szöveg jelenik meg) és egy fejléct tartalmaz. Az új file neve legyen infob.c.

Legelső lépés mint mindig:

```
#include <forms.h>
```

Aztán egyből kezdhetjük a formst, melyet most belezünk egy eljárás testébe:

```
FL_FORM *info_browser;
```

```
void p_create_form_info_browser(void)
```

```
{
    FL_OBJECT *obj;
    FL_OBJECT *ende_but;
```

```
    info_browser = fl_bgn_form(FL_NO_BOX,420,320);
```

Most, hogy elkészítettük az alapablakot, színezzük ki a hátteret, és írjuk ki a fejléct:

```
#define STR_INFO_TITLE "Információk az X-Amőbáról"
    obj =
```

```
fl_add_box(FL_FLAT_BOX,0,0,420,320,STR_INFO_TITLE);
    fl_set_object_color(obj,FL_SLATEBLUE,FL_COL1);
    fl_set_object_lcol(obj,FL_YELLOW);
    fl_set_object_lsize(obj,FL_HUGE_SIZE);
```

```
fl_set_object_lalign(obj,FL_ALIGN_TOP|FL_ALIGN_INSIDE);
```

```
fl_set_object_lstyle(obj,
    FL_BOLDITALIC_STYLE+FL_SHADOW_STYLE);
```

```
fl_set_object_resize(obj, FL_RESIZE_NONE);
```

Ezek után hozzuk létre a megszokott módon a nyomógombot, amellyel elhagyhatjuk majd a dialógusablakunkat:

```
ende_but = fl_add_button(FL_NORMAL_BUTTON,165,280,90,30,STR_BUTTON_CLOSE);
```

```
fl_set_object_lsize(ende_but,FL_NORMAL_SIZE);
```

Most már van hátterünk, fejlécünk és nyomógombunk is. Már csak a listázó menü hiányzik, amely teljesen új objektum – olyan, amilyenel eddig még nem találkozunk. Mindezek ellenére a létrehozása ismerős lesz, éppúgy mint a tulajdonságainak beállítása is:

```
obj = fl_add_browser(FL_NORMAL_BROWSER,10,40,400,230,"");
```

```
fl_set_object_color(obj,FL_WHEAT,FL_COL1);
```

```
fl_set_object_lcol(obj,FL_BLACK);
```

```
fl_set_object_lsize(obj,FL_HUGE_SIZE);
```

Tehát létrehoztuk a listázó objektumot, beállítottuk az előtér és háttér színeit, valamint a benne megjelenő betűk színét és méretét. Most már csak a listát kell beletöltenünk, amit egy file-ból fogunk tenni. Ez a file egyelőre abban a könyvtárban található, ahol a program is. Az INFO\_TXT\_FILE-t, vagyis az elérési útvonalat majd a fordításkor fordítási opcióként adjuk át:

```
#define STR_INFO_FILE "OLVASS.EL"
```

```
char st[64];
```

```
strcpy(st, INFO_TXT_FILE);
```

```
strcat(st, STR_INFO_FILE);
```

```
if (! fl_load_browser(obj, st) )
```

```
    fl_add_browser_line(obj, STR_NO_SUCH_FILE);
```

Ezzel be is fejeztük dialógusablakunk létrehozását, vagyis:

```
fl_end_form();
```

```
fl_show_form( info_browser, FL_PLACE_MOUSE,
    FL_TRANSIENT, MY_PRGNAME );
```

A második sorban azt definiáljuk, hogy az ablak az egérmutató koordinátái környékén jelenjen meg (FL\_PLACE\_MOUSE).

Ezek után már csak azt kell figyelnünk, hogy be szeretné-e fejezni a felhasználó a dialógusablakot:

```
do
```

```
{
    obj = fl_do_forms();
```

```
if ( obj == ende_but )
```

```
{
    fl_hide_form( info_browser );
```

```
    fl_free_form( info_browser );
```



```

return;
};
} while (1);

```

Mindehhez készítsünk egy header file-t `infob.h` néven, és exportáljuk ki az egész eljárást. Ezt a header file-t a főprogramba beincludolva, a következőt írhatjuk:

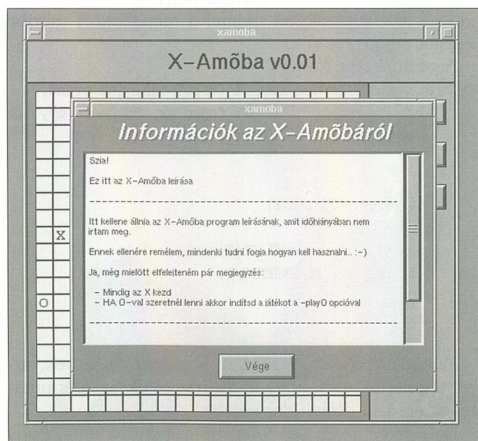
```

#include "infob.h"

if ( obj == info_gomb )
{
    p_create_form_info_browser();
    continue;
}

```

Vagyis ha megnyomják a főprogramban az [Ismerető] nyomógombot, akkor jelenjen meg a dialógusablak, és mindaddig maradjon a vezérlés a dialógusablaknál, míg ottan ki nem lépett a játékos (`xamoba_05.c`).

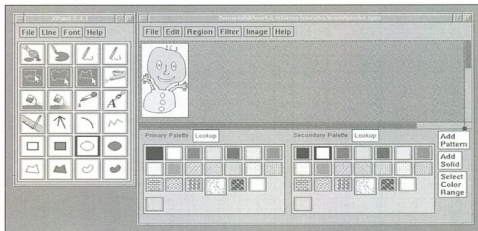


## Díszítsuk programunkat grafikával

Egy program akkor igazán szép, ha grafikus elemeket is ötvözhetünk felülete kialakításakor a már meglévő elemekkel. Szerencsére az `xforms` létrehozói ebben is segítségünkre vannak. Grafikákat kétféle formátumban (bitmap és pixmap) használhatunk programjainkban. Egy grafikát önmagában vagy akár egy nyomógombra helyezve is megjeleníthetünk.

Mivel a játékprogramnak csak három nyomógombja van, a jobb oldalon egy nagy üres terület szabadon áll. Erre a területre rakjunk most egy olyan grafikát, melyre

kattintva ugyanazt a hatást érhetjük el, mintha az [Ismerető] gombra kattintottunk volna.



Ehhez először is hozunk létre egy pixmapet. Ezt leg-egyszerűbben az `xpaint` programmal tehetjük. A meg-rajzolt képet mentsük ki `amoba.xpm` néven `xpm` for-mátumban. Az így létrehozott `pximapet` egyszerűen be-tudjuk `include`-olni a programunkba:

```

#include "amoba.xpm"

Ezek után hozzuk létre a nyomógombot, amelyre
majd később ráhelyezzük a grafikát:
FL_OBJECT *xpm_gomb;
xpm_gomb=fl_add_pixmapbutton( FL_NORMAL_BUTTON,
MY_WIN_X-96, MY_WIN_Y-166,
92, 162, "" );

```

Most már csak a `pixmap`-et kell hozzárendelni:

```

fl_set_pixmapbutton_data( xpm_gomb, amoba_xpm );

```

Már készen is van a grafikával ellátott nyomógomb. Most már csak a megnyomását kell figyelniünk a prog-ramból. Ezt legegyszerűbben ugyanazon a helyen te-hetjük meg, ahol az [Ismerető] gombot is figyeljük, mi-vel ugyanazt a funkciót szeretnénk ezzel a grafikával is ellátni:

```

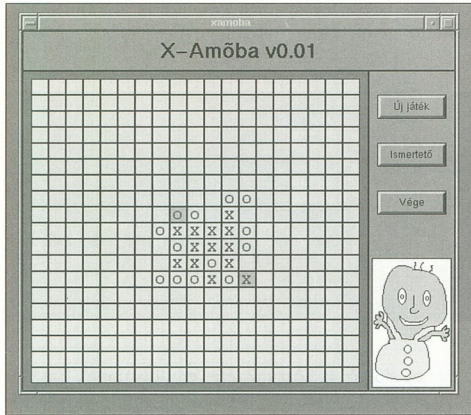
if ( (obj==info_gomb) || (obj==xpm_gomb) )
{
    p_create_form_info_browser();
    continue;
}

```

Ezek után a játékot újra lefordítva és elindítva jobb oldalt alul megtalálhatjuk a rajzunkat (`xamoba_06.c`), melyre rákattintva előtűnik az információs dialógablak.

## Változtassuk meg az egérmutatót

Van néhány olyan funkciója az `xforms`-ban található ob-jektumoknak, melyeket tőlünk teljesen függetlenül ellát-nak, és melyek végül is számunkra nem túl fon-tosak. Eme funkciók közé tartozik például, hogy ha az



egérmutatót ráhúzzuk egy nyomógombra, akkor a nyomógomb színe megváltozik, mutatva, hogy most egy egérkattintással az a kapcsoló fog aktiválódni. Ugyanakkor ha lehúzzuk az egérmutatót a kapcsolóról, akkor a nyomógomb színe visszaváltozik az eredetire. Ezt úgy érik el az xformst létrehozók, hogy a `fl_do_forms()` eljárás ténylegesen csak akkor tér vissza a programunkba, ha olyan esemény történt, melyet ténylegesen nekünk, az xformst használóknak kell lekezelnünk. De mint ahogy azt a példa is mutatja, más események is lejártszódnak a háttérben, melyekkel nekünk nem kell törődnünk. Eme más eventek feldolgozására nyújt az xforms számunkra lehetőséget az úgynevezett callback eljárások segítségével. Normális esetben az eventek közvetlenül az xforms objektumainak eseményfeldolgozó rutinjaihoz kerülnek. A callback rutinok olyan, az eventek útvonalaiba beékelte procedurák, melyek alapdefiníció szerint egyszerűen továbbadják a hozzájuk beérkező eventeket az eventfeldolgozó rutinoknak. Nekünk viszont lehetőségünk van e callback rutinok átdefinálására, és így az eventek szűrésére, illetve újabb eventfeldolgozó rutinok és eljárások beillesztésére.

Nézzünk egy példát! Azt szeretnénk, hogy ha az egérmutató az előzőleg beillesztett grafika fölé ér, akkor változzon meg az egérmutató. Ehhez először is írjuk meg a callback rutinokat. Két fajtát, mégpedig azért, mert az eventek is több csoportra vannak osztva, és egy nyomógombnál vannak olyan eventek, melyek a nyomógombra való érkezéskor aktiválódnak, és vannak olyanok, melyek a nyomógomb elhagyásakor generálódnak. Ezeket két különböző callback rutinban kell figyelnünk (prehandler és posthandler).

Lássuk először a prehandler-t:

```
int p_pre_handler_xpm( FL_OBJECT *ob, int event,
                      FL_Coord mx, FL_Coord my, int
key, void *xev)
{
    if (event==FL_ENTER)
    {
        fl_set_cursor(FL_ObjWin(ob), XC_question_arrow
);
        return( FL_PREEMPT );
    }
    return(0);
}
```

Vagyis ha az egérmutatóval az objektum területére belépünk, akkor változzon az egérmutató kérdőjeles nyílá. Az utána visszaadott érték (`FL_PREEMPT`) azt jelenti, hogy további feldolgozásra az event ne kerüljön. Minden más eventet érintetlenül továbbadunk a 0 érték visszaadásával, ami azt jelenti, hogy az eventeket az xforms eventfeldolgozó rutinjai a megszokott módon kapják meg.

A posthandler:

```
int p_post_handler_xpm( FL_OBJECT *ob, int event,
                       FL_Coord mx, FL_Coord my, int
key, void *xev)
{
    if (event==FL_LEAVE)
    {
        fl_set_cursor(FL_ObjWin(ob), -1 );
        return ( FL_PREEMPT );
    }
    return(0);
}
```

Működési módja ugyanaz, mint a prehandlernek, azal a különbséggel, hogy itt visszaállítjuk az egérmutatót az eredeti (default) formájába.

Ha készen van a két rutin, meg kell mondani, hogy melyik objektumnál kerüljenek felhasználásra:

```
fl_set_object_prehandler( xpm_gomb,
                          p_pre_handler_xpm );
fl_set_object_posthandler( xpm_gomb,
                           p_post_handler_xpm );
```

Ezek után a programot újrafordítva az egérmutató a grafika területén egy kérdőjel formáját fogja felvenni, míg a grafikából kihúzva eredeti formájára (egy nyíl) változik vissza (`xamoba_07.c`).

Abban a reményben adom közre eme kis írományom, hogy páran kedvet kapnak majd a cikk nyomán az X-rendszere való programozásra, és a közeljövőben sok szép új programmal találkozhatunk majd. ■

**VTCD VIDEOTON**  
Kompaktlemez-gyártó Kft.

Székesfehérvár, Aszalvölgyi u. 1.  
☒ 8001 Székesfehérvár, Pf. 175  
☎ (06-22)329132  
Fax:(06-22)329133

COMPACT  
disc  
DIGITAL AUDIO

COMPACT  
disc+  
DIGITAL AUDIO

COMPACT  
disc  
DIGITAL VIDEO

COMPACT  
disc  
Interactive

COMPACT  
disc



✓ *Kompaktlemez*

✓ *Kompakt Technológia*

✓ *Kompakt Szolgáltatás*

# Nem csak gyűjtőknek

**ALBACOMP**  
*activa*

MMX™ TECHNOLÓGIÁVAL  
ELLÁTOTT  
PENTIUM® PROCESSZOR



Albacomp Rt.  
8000 Székesfehérvár  
Hosszú sétány 4-6.  
Tel.: (22) \*315-414  
Fax: (22) 327-532

Budapesti Kirendeltség  
1139 Budapest  
Frangepán u. 8-10.  
Tel.: 12-91-493  
Tel/fax: 14-90-152

Szakküszletek:  
1065 Budapest  
Nagymező utca 25.  
Tel.: 11-18-095, 13-18-108  
Fax: 13-18-108

1011 Budapest  
Fő utca 31.  
Tel.: 201-4409  
Fax: 201-4322

3525 Miskolc  
Széchenyi u. 49.  
Tel.: (46) 354-266  
Tel./fax: (46) 353-100

