

nálók, de nem a szakemberek számára. Megpróbálkozhatunk a „szolgáltatáselrejtés” alkalmazásával, amit én a Microsoft Word 2000-ben láttam először. A szakembereknek szánt programcsomagok viszont valósággal kérkednek az elérhető szolgáltatások százaival. Vessünk például egy pillantást a *Blender* nevű 3D-modellezőre, amely egyszerre több tucat nyomógombot rak ki a képernyőre, és ezek a gombok az üzemmódtól függően állandóan változnak, mindig újabbak bukannak elő!



Hiába grafikusak ezek a felületek, igazából nem lehet őket felhasználóbarátnak nevezni, hiszen előtanulmányok nélkül a zöldfülűek semmit sem tudnak kezdeni velük. Az ilyen programokat azonban az irodai programcsomagokkal ellentétben nem akarják mindenkinek eladni, beleértve az olvasni tudó kisdedeket és a homályosodó szemű aggasztányokat. A szakembereknek fejlesztő programozók számára tehát nem az a kérdés, hogy megmutassák-e a programban rejlő lehetőségeket, hanem az, hogy miképpen tegyék meg. Általánosan elfogadott módszer, hogy a szolgáltatásokat ilyen-olyan szempontok szerint csoportosítják, majd menükre, gyorsbillentyűkre, ikonsorokra vagy nyomógombokra fűzik fel őket. A párbeszédablakos megjelenítés kényelmes és felhasználóbarát, ha ritkán használt beállításokról van szó, de igen hátráltatja a munkát, ha gyakorta ismétlődő parancsok előhívására használjuk. Gondoljunk bele például, hogy a kijelölt szövegrész kivágásakor mennyire időigényes mozgássor az F10 funkcióbillentyű megnyomása, majd a kivágás menütétel kiválasztása a NYÍL billentyűkkel, és végezetül az ENTER leütése. Gyorsabb, ha a kivágás ikont nyomjuk meg gépeléskor, de még ilyenkor is oda kell vinnünk a kezünket az oldalt lévő egérhez, ami miatt meg kell szakítanunk a szövegbevitelt. Gépeléskor tehát a gyorsbillentyű használata adja a legjobb eredményt, hiszen például az Emacsban elég megnyomni a CTRL-W billentyűkombinációt a kijelölt szövegrész kivágására, és az adatbevitel máris folytatható

anélkül, hogy a kezünket egy pillanatra is arrébb kellett volna mozdítanunk. Mivel az Emacs fejlesztői tisztában voltak ezzel az ergonómiai törvényszerűséggel, nem nagyon törték magukat, hogy más, kevésbé hatékony, de mások által felhasználóbarátnak mondott megoldásokat keressenek. Érdekes módon az XEmacs sem törekszik erre, pedig azzal a céllal hozták létre, hogy könnyebben kezelhető legyen. Elég, ha egy pillantást vetünk rá, és rögtön látjuk, hogy a fejlesztők megelégedtek néhány ikonnal (összesen 15-tel), amelyeknek a felét én személy szerint főlölegesnek tartom a mindennapi munka szempontjából. Miért kell például az Info ikont kirakni a szemünk elé, amikor nem tesz egyebet, mint megjeleníti a Sűgőt? Ezzel szemben a Microsoft Word szövegszerkesztőben nem 15 ikont, hanem 15 ikonsort találhatunk! Az XEmacsban inkább a menük burjánzanak. Mégsem állítanám, hogy az XEmacs fejlesztői lusták lettek volna, amikor kispórolták az ikonokat. Inkább azt feltételezem, hogy nem látták értelmét annak, hogy ikonokkal vagy nyomógombokkal zsúfolják tele az új Emacs-változatot, hiszen az a programozói réteg, amelyik az Emacs-környezetet használja, nem igényel ilyen változtatásokat. A fejlesztők tehát a programozás és a leíráskészítés szempontjából a leghatékonyabb megoldást választották, azaz a gyorsbillentyűkkel való parancshívást, még akkor is, ha ez a kezdők számára első pillantásra elrettentőnek tűnik. Nincs királyi út! Legfeljebb az Emacs.

Az Emacs és az Xemacs

A kétfajta Emacs között nincs nagy különbség, de hamar észrevehetjük, hogy nem teljesen egyforma a kettő. Az XEmacsban hiába kerestem a szöveges állományokat átalakítás nélkül beolvasó `find-file-literally` parancsot, csak az Emacsban találtam meg. Bizonyos billentyűkombinációk is másként működnek a két változatban, de összességében nem nagyok az eltérések. Tanulás közben célszerű mindkettőt kipróbálni, és végül annál maradni, amelyik jobban tetszik nekünk. Manapság a legtöbb alkalmazást valamilyen grafikus felhasználói felületre írják, ezért felmerülhet a kérdés, hogy van-e értelme az X Window nélkül valamelyik terminálról indítani az Emacsot, hiszen mindegyik ablakkezelőben van terminálemulátor, és abban már futhat az Emacs. De mi tehet az a programozó, aki éppenséggel egy ablakkezelő megí-

rásába fog bele? Neki valószínűleg a parancssorról kényelmes tesztelnie új programját, hiszen éppen fejlesztés alatt álló ablakkezelője feltehetően még nem igazán használható. Ilyenkor csak valamelyik parancssori szerkesztőt futtathatja.

Ha nincs egerünk, akkor a terminálon indított Emacs még akkor is új élményt fog jelenteni számunkra, ha már jártasak vagyunk a terminálemulátorban futtatott Emacs használatában. Megszoktuk már, hogy a legtöbb alkalmazásban az F10 nyitja le a menüt, és nincs ez másként az Emacsban sem. A parancssorra tévedt felhasználó azonban meglepetten tapasztalja, hogy hiába nyomkodja az F10 funkcióbillentyűt, semmi sem változik a menüsorban. Minél feszültebben figyel azonban a képernyő felső részére, annál kevesebb esélye lesz arra, hogy észrevegye, mi történik az alsó fertályban. Az Emacs ugyanis a képernyő alján, a mini átmeneti tárbán (pontosabban a visszhangterületen) írja ki a menüket, és ott is kell választanunk a megjelenő menütelemek a balra vagy jobbra nyílak közül, majd az ENTER megnyomásával.

Hasonló furcsaságokkal bármikor találkozhatunk az Emacsnál, de ha már megismertük, többé nem fogunk meglepődni.

Utószó

Nyilvánvaló, hogy a vi vagy az Emacs más, mint amit a többi operációs rendszerekben megszoktunk, de az a tény, hogy furcsa és szokatlan, még nem lehet a minősítés alapja. A szövegben szereplő kódbetűs szavak nemcsak a programok nevére utalnak, hanem megegyeznek a héjba beírandó programindító parancsokkal, a nagybetűvel kezdődő szavak viszont csak a programok neveit jelölik. A következő részben röviden ismertetem az Emacs használatát.



Szaló István

(ratiosoft@freemail.hu) tanár, immár több mint másfél évtizede foglalkozik programozással, de csak a Java és a Linux

megismerése után tudta meg, hogy mi is az igazi programozás. Több írása megjelent már a hazai számítástechnikai lapokban. Ha néha feláll számítógépe mellől, rendszerint művészettörténész feleségével és kisiskolás lányával „találja szemben” magát.



Fordítsunk rendszermagot!

A nyílt operációs rendszerek egyik legnagyobb előnye, hogy a rendszermagot a felhasználó is bármikor újrafordíthatja, ezáltal csökkentheti a méretét és gyorsíthatja a rendszer futását.

A rendszermag újrafordítása a Linux világában ugyan hétköznapi műveletnek számít, mégis számos nehézséget okozhat a kezdő felhasználók körében. Cikkünk nekik próbál segítséget nyújtani. Bármely operációs rendszer legfontosabb része a rendszermag (kernel), amelynek legfontosabb feladata a felhasználói programok és a gépünk közötti kapcsolattartás biztosítása. Ezzel azonban még nem ért véget a tevékenységi köre, ugyanis szintén a rendszermag felelőssége a folyamatok (process) futásának felügyelete, továbbá a különböző biztonsági szabályok betartatása (nem engedi, hogy egy futó alkalmazás egy másik program által használt memóriaterületre írjon stb.).

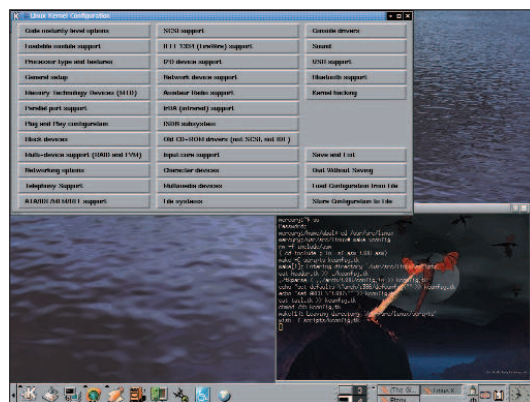
A nyílt forráskódú rendszereknél a rendszermag forráskódjához is bárki hozzáférhet, ezért a felhasználók saját kezűleg fordíthatják újra, ha akarják. Milyen előnyökkel járhat a rendszermag újrafordítása? Először is várhatóan gyorsabb lesz, mint az előre lefordított „gyári” rendszermag, ugyanis a fordítóprogram a mi processzorunkhoz hangolja. Hasznos az is, hogy a fordítás előtt a felhasználónak lehetősége nyílik kiválasztani, mely összetevők szerepeljenek, illetve ne szerepeljenek a kész rendszermagban. Amennyiben például nincs és várhatóan nem is lesz semmiféle SCSI-eszközünk, a lefordított rendszermagból a teljes SCSI-támogatást ki lehet hagyni. Ezáltal jelentős méretbeli csökkenést érhetünk el. A rendszermag újrafordításával tehát elkészíthetjük a kifejezetten saját gépünkhöz illeszkedő rendszermagot.

A Windows-hoz vagy OS/2-höz szokott felhasználók számára minden bizonnyal furcsán hangozhat, hogy a nyílt forrású operációs rendszerek világában a rendszermag újrafordítása teljesen hétköznapi műveletnek számít. Maga a fordítás nem nevezhető bonyolult műveletnek, sarkpontja egyedül az összetevők kiválasztása, amit azonban részletesebben is be fogunk mutatni. A rendszermag

fordítása egyébként semmiféle programozói előképzettséget nem igényel. Kezdjük is neki! Legelső feladatunk magának a rendszermag forráskódjának a beszerzése lesz. A forrást Linuxunk telepítő CD-jén is megtalálhatjuk, de a legújabbat mindig fellelhetjük a CD-mellékletben, vagy letölthetjük az `ftp://ftp.kernel.org`-ról, illetve annak magyar tükréről, az `ftp://ftp.hu.kernel.org` címről. Jelen pillanatban a 2.2.x-es és a 2.4.x-es sorozatot párhuzamosan fejlesztik, e cikk írásának pillanatában az előzőből a 19-es, az utóbbiból pedig a 14-es a legfrissebb. Itt szeretnénk felhívni a figyelmet, hogyha régebbi Linux-változattal rendelkezünk, amely még a 2.2-es sorozatra épül, és át szeretnénk térni a 2.4-es rendszermagra, akkor bizonyos csomagokat frissítenünk kell. Ilyen például a telefonos interneteléshez szükséges PPP-démon vagy a modulokat kezelő `modutils` (lásd később). E csomagok listáját a rendszermag leírásában találhatjuk meg. A frissítéseket Linux-változatunk hivatalos honlapjáról is letölthetjük. A magforrás becsomagolva körülbelül 22 MB, a lefordításához azonban körülbelül 60–80 MB szabad tárhelyre lesz szükségünk. Az Interneten elérhető rendszermagforrásokat általában `gzip`-vel vagy `bzip2`-vel tömörítik: az első esetben a kiterjesztés `tar.gz`, a másodikban pedig `tar.bz2`.

A forrást másoljuk a `/usr/src` könyvtárba. Ha `gzip`-vel tömörítették, akkor a `gzip -d kernel-xxx.tar.gz` paranccsal csomagolhatjuk ki, `bzip2` esetében a `bunzip2 kernel-xxx.tar.bz2` utasítást használhatjuk. (az `xxx` a rendszermagnak megfelelő változatszám) Ezt követően a létrejött `.tar` kiterjesztésű állomány kicsomagolásához adjuk ki a `tar -xvf kernel-xxx.tar` parancsot. Ez a parancs létrehoz egy Linux nevű könyvtárat. Ha már

található ilyen könyvtár az adott helyen, előtte nevezzük át, például `Linux-old-ra` (mv `Linux Linux-old`). Amennyiben mindent jól csináltunk, a forrást magát az újonnan létrejött `linux` könyvtárban találhatjuk meg. A tar-állományra a továbbiakban



nem lesz szükségünk, tehát nyugodtan letörölhetjük, ne foglalja fölöslegesen a helyet.

Most lépünk be a magforrást tartalmazó `linux` könyvtárba! A rendszermag fordításának első és egyben legnehezebb lépése a kész rendszermag összetevőinek a kiválasztása. Mielőtt buzgón hozzálátnánk, meg kell beszélnünk egy fontos dolgot.

A Linux-rendszermag fontos tulajdonsága, hogy modularizált. Ez azt jelenti, hogy bizonyos alkatrészek és szolgáltatások támogatását nem feltétlenül kell közvetlenül a rendszermagba fordítanunk, hanem lehetőségünk nyílik rá, hogy modulokat készítsünk. Ezeket a modulokat bármikor kedvünkre betölthetjük a memóriába, illetve amennyiben feleslegessé váltak, el is távolíthatjuk őket onnan. Miért jó ez nekünk? Bizonyára akadnak olyan eszközeink, amelyeket nem használunk állandóan, a legjobb példa erre talán a hangkártya. A hangkártya szolgáltatásaira a rendszer mindennapi használatában nincs szükségünk, csupán abban az esetben, ha zenét akarunk hallgatni vagy lazítás-

képp valamely linuxos játékkal szeretnénk egy kicsit játszani. Ha jobban megfontoljuk, beláthatjuk, hogy teljesen felesleges a hangkártyatámogatást a rendszermagban „tárolnunk”, sokkal célszerűbb, ha modulba tesszük. Ennek köszönhetően a „hangos eszköz”-támogatás csak akkor kerül a memóriába, amikor hangkártyánkat ténylegesen „dalra fakasztjuk”.

Érdeemes minél jobban kihasználnunk a Linux-rendszermag eme előnyét. A bevált szokás az, hogy csak azokat az elemeket fordítjuk közvetlenül a rendszermagba, amelyekre a rendszer elindításához feltétlenül szükség van. Nem érdemes modulba tenni azoknak az egységeknek a támogatását, amelyeket a rendszer futása közben állandóan használunk: ilyen lehet például egy hálózati kiszolgáló esetében a hálókártya. Amelyik támogatást csak lehet, mind „dobáljuk” modulba.

A modulokat egyébként a `modprobe` parancs segítségével tölthetjük be, a feleslegessé vált modulok memóriából való eltávolítására pedig az `rmmod` utasítás szolgál. A betöltött modulokat az `lsmod` parancs listázhatjuk ki. Egyes modulok betöltésekor értékeket is meg kell adnunk, hogy hol és mit, arról a rendszermag leírásában olvashatunk bővebben. A legtöbb modul betöltéséről azonban a rendszermag saját maga gondoskodik.

Lássunk neki az elemek kiválasztásának! Ehhez többféle út is kínálkozik: az egyik a konzolos menüvezérelt alkalmazás, amelyet a `/usr/src/linux` könyvtárból a `make menuconfig` utasítás segítségével kelthetünk életre. Akik a grafikus környezetet kedvelik jobban, azok egy grafikus konzolból adják ki a `make xconfig` parancsot. A Linux-rendszermag fejlesztői az önsanyargatókról sem feledkeztek el: a `make config` parancsot az ő figyelmükbe ajánljuk.

A különböző elemeket különböző osztályokba csoportosítva találjuk. A továbbiakban helyhiány miatt csak a legfontosabbakra térhetünk ki (az összes támogatás részletes bemutatására a fél újság sem lenne elég). Mindenesetre bővebb tájékoztatásért nézzük át a rendszermag leírását vagy nyomjuk meg a **Help** gombot.

Code maturity level options

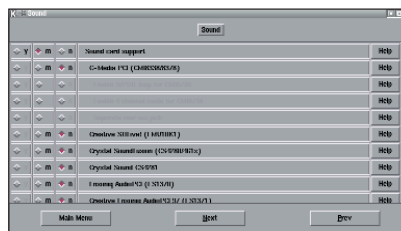
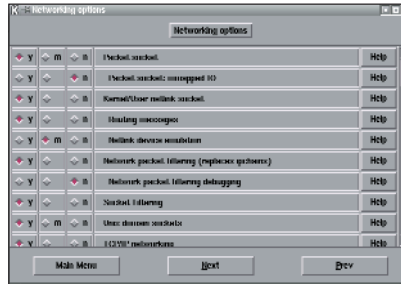
Ha bekapcsoljuk, elérhetjük a rendszermagban szereplő, de még csak kísérleti (EXPERIMENTAL) állapotban lévő támogatásokat is. Figyelem, az ilyen szolgáltatások nem üzembiztosak, ezért mindenki csak a saját felelőségére használja őket!

Loadable module support

Az **Enable loadable module support**-ot mindenképpen fordítsuk be a rendszermagba, mivel nélküle nem élvezhetjük a modulok nyújtotta előnyöket.

Kernel module loader

A modulok önműködő betöltését teszi lehetővé.



Processor type and features

Itt adhatjuk meg processzorunk típusát. Ez azért fontos, mert a fordítóprogram a rendszermagot erre a processzorra fogja hangolni, ennek köszönhetően rendszerünk sebessége jelentősen nőhet.

High memory support

Ezt állítsuk **Off**-ra, amennyiben 1 GB-nál kevesebb fizikai memóriával rendelkezünk.

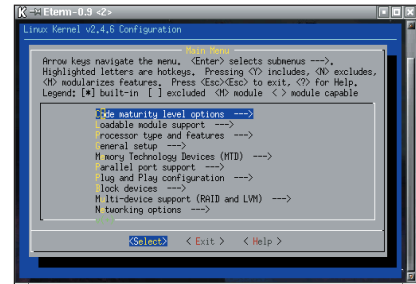
Symmetric multi-processing support (SMP)

Csak abban az esetben kapcsoljuk be, ha egynél több processzorra bírnunk, ugyanis ez a támogatás teszi lehetővé, hogy a rendszermag egynél több processzort használjon a folyamatok futtatására.

General setup

A **PCI access mode**-nál adhatjuk meg, hogy a rendszermag milyen módon keresse meg a különböző PCI-os eszközöket. Ez történhet a BIOS segítségével, de közvetlenül (**Direct**) is. A legbiztosabb, ha az **Any**-t választjuk; ebben az esetben a rendszermag PCI-os eszközeinket először a BIOS-on keresztül, majd ha ez sikertelen, közvetlenül próbálja meg elérni. Az MCA-támogatás a PS2-es eszközök meghajtására szolgál. Ha PCMCIA-s kártyákkal is rendelkezünk, azok támogatását is itt kapcsolhatjuk be. A **System V IPC**-t feltétlenül tegyük működévé,

ugyanis ez teszi lehetővé a futó folyamatok közötti kapcsolattartást. Nagyon ügyeljünk rá, hogy a **Support for ELF binaries**-t mindenképp magába a rendszermagba fordítsuk (és ne modulba!). Az elf a Linux futtatható binárisainak a formátuma, olyasmi, mint a Windows világában az exe. A többi bináris támogatását viszont nyugodtan modulba is helyezhetjük. Ha mindenánnak energiaellátását programból is szeretnénk szabályozni, ne felejtjük el bekapcsolni a **Power management support**-ot sem!



Parallel port support

Ez a párhuzamos kapu támogatása, nyugodtan tegyük modulba. Figyelem, amennyiben PC-t használunk, a **PC style hardware**-re is szükségünk lesz!

Block devices

Ide tartozik a PC-s hajlékonylemez-meghajtók támogatása, amit nyugodtan tegyük modulba, még abban az esetben is, ha Linuxunkat lemezről indítjuk. A rendszermagot ugyanis a lemezről a Linux indításvezérlője, a LILO tölti be, ami saját maga is tudja a lemezt, illetve a merevlemezeket kezelni.

Multi-device support

Ennél a menüpontnál a RAID-eszközök különböző támogatásait találjuk. A **linear** azt jelenti, hogy az összekötött lemezeket folyamatosan, egymás után töltjük meg. Az is megoldható, hogy több lemezt lássunk egy fájlrendszerként, ilyenkor a **striping** (csíkozás) nyújthat hasznos szolgáltatásokat. A **mirroring**-gal (tükrözés) pedig megoldhatjuk, hogy két különálló lemezre pontosan ugyanazokat az adatokat írjuk fel. Ez nagymértékben növeli adataink biztonságát, ha ugyanis az egyik lemez megsérül, adataink a másikon még mindig elérhetőek lesznek. Az **LVM support** lehetővé teszi, hogy lemezterületeink összevonásával logikai kötetet hozunk létre. Ezt a szolgáltatást csak a 2.4-es rendszermagok tartalmazzák. Akkor tehet jó szolgálatot, ha hirtelen lenne sok helyre szükségünk, de nem akarunk RAID-et használni.