

ga, egy ilyen gépet véletlenül se bízunk olyan emberre, akinek nincs kellő tapasztalata, és nagyon fontos, hogy először élesszük fel rajta a szolgáltatásokat, alakítsuk ki a biztonsági rendet, működjön rendesen a levelezés, a fax, a fájlkiszolgálás, a nyomtatás, az internetkapcsolat és az útvalasztás. Csak a szolgáltatások üzembiztos működése után kezdjük meg az átállást (nyugi, utána is lesz elég bajunk).

Az új rendszer üzembe helyezése szinte biztos, hogy az összes munkagép átállítását magával vonja (még ha DHCP-t használunk, akkor is elképzelhető). Számoljunk elég idővel, hiszen vagy minden felhasználónak végig kell mesélnünk a történetet, vagy készítenünk kell egy leírást, amelyben részletezzük, hogy mi és miért történik. A levélírással hatnyolc ügyfélgép fölött biztos, hogy időt takarítunk meg. Ha ügyesek vagyunk és bízunk a felhasználókban, a leírásban azt is leírhatjuk, hogy miként tudják saját maguk elősegíteni az átállást, és hogy milyen lépésekben és határidőkkel történik meg a váltás, valamint hogy milyen úton jelezzék, ha különleges igényük van.

Most jön a legfárasztóbb rész: a szakaszokat egyesével átállítjuk, az ügyfelek-nél költözés, majd a régi szolgáltatások tiltása az adott szakaszra. Hogy miért fontos tiltani a régi szolgáltatásokat?

Hogy nehogy nagyot koppanjunk, amikor (pár nap vagy hét) múlva nagy nyugalommal ízekre szedjük a régi gépet, a dolgozók pedig dörömbölnek, hogy nem tudnak dolgozni. Miután az első kis csoport átköltözött, adjunk nekik időt, hogy minden szolgáltatást élesben is kipróbáljanak. Ez órákat, napokat, de előfordulhat, hogy heteket jelent. Miután az első szakasz átállt és gond nélkül dolgoznak az új gépen, jön a tisztogatás. Ehhez szükséged lesz egy csavarhúzóra, kötéldegzetre és a főnök előzetes meggyőzésére. Ugyanis most jön, hogy az új szakasz összes gépéből ki kell követelni a modemeket, majd mindenkinek el kell magyarázni, hogy csak a központi gépen tárolja a fontos adatokat (mert az új rendszer legalábbis tükrözött lemezekkel dolgozik), hogy ne merjen hülyeségeket telepíteni (ismerek olyat is, aki a hajlékonylemezes és a CD-meghajtót is kisereli), hogy véletlenül se osszon meg semmit a gépből (jellemzően a C: van megosztva, jelszó nélkül), hogy milyen levelezőt használjon és így tovább. Ez az a pont, amikor a rendszergazda még a főnökség tagjait is megelőzi a körözési listákon. Ezután csendben a háttérbe vonulunk és figyelünk. Ha minden felmerült kérdést megoldottunk, túl vagyunk a nehezén. Most már csupán ugyanezt kell végigzongoráznunk az összes szakaszon.

Miután az összes gép átállt és a régi kiszolgáló elérését mindenhol letiltottuk, még érdemes egy ideig – a biztonság kedvéért – működésképes állapotban tartani, és később, amikor már senki sem keresi, ráérünk szétszedni.

Igyekeztem röviden összefoglalni egy átállás menetét, természetesen minden környezet más és más, lehet, hogy több kiszolgáló van, előfordulhat, hogy az új gép csak különböző feladatokat vesz át, vagy a munkagépeket egytől egyig újra kell telepíteni... a változatok száma végtelen. Egy viszont biztos, ha alaposan átgondoljuk az átállást, nagyon sok keserűségtől kíméljük meg magunkat. És még egy gondolat: vegyünk legalább egy jó könyvet, ami a biztonság alapjaival foglalkozik és az összes felhasználóval olvastassuk el! Könnyebb a dolgunk, ha nem nekünk kell ötvenszer elmagyarázni, hogy miért ne használjon a munkatárs olyan levelezőt, ami önműködően elindítja a levelekben lévő vírusokat!



Szy György
a Linuxvilág főszerkesztője, a Kiskapu Kiadó vezetője.
Mindenki véleményét és levelét örömmel

várja az alábbi levélcímen:
Szy.Gyorgy@Linuxvilag.hu

© Kiskapu Kft. Minden jog fenntartva

Hazai linuxos cégek gyűjteménye

Sokszor keresnek meg minket, hogy ismerünk-e jó rendszerfelügyelő céget, tudunk-e valakit, aki vállalja egy kiszolgáló telepítését, hol lehet linuxos tanfolyamokra jelentkezni, kihez fordulhatnak ügyesbajos kérdéseikkel stb. Mivel mi magunk sem ismerjük a hazai linuxos közösség minden tagját, arra gondoltunk, hogy indítunk egy sorozatot, amelyben mindig két-három, az adott területen dolgozó céget mutatunk be – a hálózatépítőtől a rendszerfelügyelőn át a programozóig. Igen ám, de ha ilyen rendszert szeretnénk indítani, jó volna, ha a cégeket egy pontos rendszerbe tudnánk beilleszteni. E rendszer kialakításához kérem most a segítségeteket.

Mit érdemes, és mit kötelező leírni egy ilyen cégről? Természetesen fontos, hogy a cég mikor alakult, hogy mióta foglalkozik Linuxszal, milyen típusú rendszerekkel dolgozik, hol található a székhelye, milyen ügyfélkört céloz meg. Emellett szerintem rendkívül fontos, hogy az adott cégen belül hány munkatárs ért az adott feladathoz, például egy rendszerfelügyeletet vállaló cégnél hány rendszergazda dolgozik, a cég mennyire van leterhelve, vállal-e 24 órás vagy hétvégi ügyeletet, gond esetén milyen határidővel dolgoznak, és természetesen milyen árkategóriában.

Az ár nehéz kérdés. Lehetetlen megmondani, hogy valaki mennyiért

telepít egy rendszert, amíg nem tudja pontosan, mik is az igények. Az sem mindegy, hogy az adott munka elvégzése után hogyan adja át a készterméket, például egy gép szállítása után a cég vállalja-e, hogy amennyiben az általa összerakott gép valamelyik alkatrésze nem (vagy nem megfelelően) működik Linux alatt, azt kicseréli? Emellett egy vezető számára az is fontos, hogy a cég referenciákat tudjon adni. Sőt, az volna a legjobb, ha néhányuk véleményét ki lehetne kérni. Szeretném, ha létre tudnék hozni egy olyan listát, amely alapján az érdeklődő könnyen ki tudna választani mondjuk egy megbízható, elfogadható áron dolgozó rendszerfelügyelettel foglalkozó céget, vagy például egy komoly tapasztalatokkal rendelkező hálózatépítő társaságot.

Mivel mi ebben csak katalizátorszerepet játszhatunk, kérek mindenkit, írja meg nekem véleményét, a gondolatait, hogy milyen cégekre kíváncsi, milyen kérdéseket tenne fel, milyen szerkezetben látná örömmel a listát. Egyúttal ez felhívás a hazai linuxos cégek és vállalkozók felé is! Várjuk olyan társaságok és vállalkozók jelentkezését, akik ki mernek állni szolgáltatásaikkal a nagyvilág elé – vállalva a megmérettetést!

Szy György



A szerkesztők és az Emacs (1. rész)

Ezt az írást nem a Linux-rendszert és a programozást jól ismerőknek szánom, hanem azoknak, akik mindkettővel most ismerkednek.

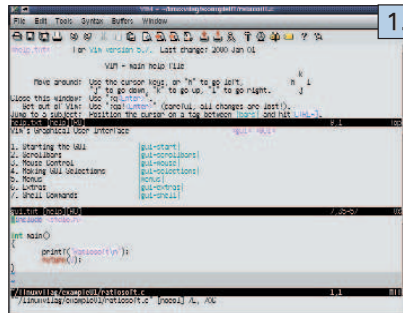
Elsőként a forráskód megírásához használható szerkesztőket mutatom be a teljesség igénye nélkül, majd az Emacs fejlesztőkörnyezettel foglalkozom hosszabban.

Szövegszerkesztők

A programokat először meg kell írni, amihez kell egy nekünk tetsző egyszerű vagy kevésbé egyszerű szövegszerkesztő, amely ASCII formátumú állományt hoz létre. Ebből következik, hogy haszontalan lenne az OpenOffice nagy tudású szerkesztőjét használni, mert fölöslegesen sok helyet foglal el a memóriában, és semmiképpen sem tudjuk kihasználni a képességeit, hiszen a programírásban nem segítenek a kifinomult formázási lehetőségek, nem célszerű sokféle betűkészletet használni, és számos programozót zavar, ha változó szélességű és nagyságú betűket lát maga előtt bogarászás közben. Ne feledjük, hogy bármilyen szépre is formázzuk meg forráskódunkat, a formázó karakterek abban a pillanatban nyom nélkül eltűnnek, amikor forráskódunkat ASCII formátumban kimentjük, a fordítók pedig csak a formázatlan állományokat hajlandók elfogadni!

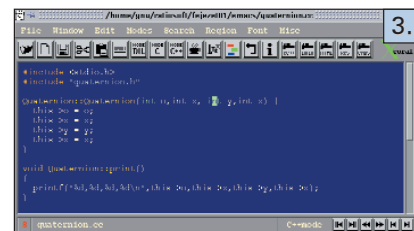
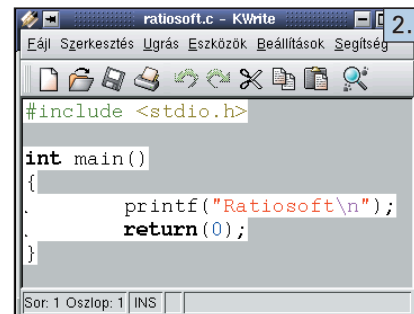
A fentiekből következik, hogy a merészebbek programozásra is használhatják a legendás híró *vi* szerkesztőt. Erről a kezdőknek azt illik tudni, hogy a kellően tájékozottak „viáj”-nak ejtik, és hogy a `:q!` paranccsal lehet kilépni belőle. Ez nem segít mindig, ilyenkor nyomjuk meg az `Esc` billentyűt, és próbálkozunk újra. Ha már fel vagyunk vértézve ennyi tudással, akkor legalább nem kell újraindítani az egész rendszert, hogy visszakaphassuk a parancssor készenléti jelét, ahogy azt egyik ismerősöm tette a *vi*-t próbálgatva. A mindenre elszántak próbálkozhatnak a *vi* többalakos változatával, az *xvi* (ejtsd: „eksz-vi-áj”) szerkesztővel, vagy az *xvi*-nál is fejlettebb *vim*-mel. A *vim* a „*vi* improved” (tökéletesített *vi*) szavak rövidítése. A *vi*-nak az elmúlt évtizedekben számos mutációja született, és ha telepítve vannak a gépünkön, kipróbálhatjuk őket. Írjuk be a parancssoron az `ed`, `ex`, `gex`, `vi`, `gvi`,

`view`, `gview`, `vim`, `gvim`, `rvim`, `rview`, `rgvim` vagy `rgview`, `elvis`, `nvi` parancsokat és az olvasni vagy szerkeszteni kívánt állomány nevét! Ezek a parancsok többnyire ugyanazt a végrehajtható állományt indítják el, de különböző bővítményekkel vagy éppen korlátozásokkal. Például a `view` vagy a `view-ra` végződő parancsok használatkor a megnyitott fájlt csak olvasni lehet, a parancsok elején lévő `r` betű pedig a „restriction”, azaz megszorítás szóra utal. A korlátozott parancsok valamilyen



módon szűkítik a *vi* lehetőségeit, például nem indíthatunk parancsokat a *vi*-ből, vagy nem függeszthetjük fel ideiglenesen a működését. A *gvim* (1. kép) a Windowsban megszokott menüket és ikonokat varázsolja elénk, és viszonylag jó leírással bír. Első pillanattól otthon fogjuk érezni magunkat benne, ha a telepítés után sikerül elindítanunk. Ha nem, próbáljuk meg a `~/gvimrc` fájl testreszabni. A *vi* általában nem alkalmas bináris állományok szerkesztésére, ehhez inkább a `hex`, `vche`, `vche-nc`, `vche-raw`, `khexedit` vagy a `ghex` programokat használjuk. Szintén kéznél van a Midnight Commander belső fájlkezelője, amely hexadecimális szerkesztő is egyben. A *vimtutor* program végigvezet bennünket a legfontosabb *vi*-parancsokon. Többek közt megtudhatjuk, hogy mely két üzemmód létezik a *vi*-ban, hogyan indíthatjuk el, hogyan mozgathatjuk a kurzort, hogyan szerkeszthetjük a fájlt, és miképpen léphetünk ki belőle. A `pico` szintén parancsori szövegszerkesztő, de könnyebb megtanulni, mint a *vi*-t. Ezt már az is mutatja, hogy

mindjárt induláskor kiírja, hogyan lehet kilépni belőle. Megemlítem még a *Joe* szerkesztőt, amit ötféle üzemmódban használhatunk. Ha a `joe` helyett a `jpico` parancsot ütjük be, akkor a *Joe* a most említett *Pico* szerkesztőt utánozza,

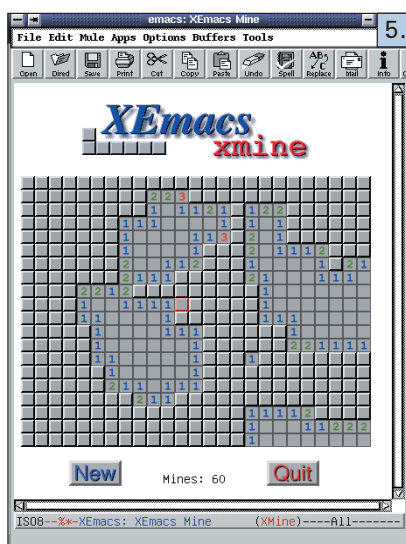
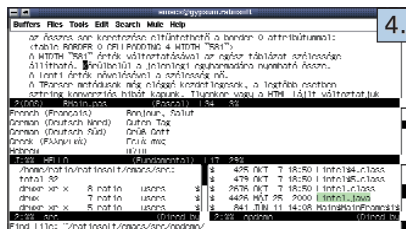


ha a `jstar` parancsot, akkor a hajdan jól ismert WordStar szövegszerkesztőt. A fenti programok mindegyike igen nagy tudású, de használatukhoz némi tapasztalatra van szükség. Az `xedit`, `kedit` és a `gedit` a Windows felől érkezők számára minden bizonnyal barátságosabbnak fognak tűnni, mint parancssori elődeik, de azt is rögtön tapasztalhatjuk, hogy ezek sem tudnak sokkal többet, mint a `notepad.exe`.

Az Emacs

A Linux-programozással foglalkozó szakkönyvek szinte kivétel nélkül a *GNU Emacs* fejlesztőkörnyezetet ajánlják. Figyeljük meg, hogy immár nem a „szerkesztő” szót használtam, hanem az IDE (Integrated Development Environment) mozaikszóra utaltam, ami magyarul összetett fejlesztőkörnyezetet jelent. Mégse gondoljunk azonban a *Linuxvilág* augusztusi számában ismertetett *KDevelop*hoz vagy a *Borland Kylix*hoz hasonló *RAD* (Rapid Applica-

tion Development, azaz gyors alkalmazásfejlesztés) eszközökre. Az Emacs a Unix-hagyományokba gyökerezik, nehézkes és barokkosan bonyolult. Némelyek szerint az Emacs gonosz, mások viszont három részre osztják a világban élő embereket: azokra, akik Emacsot használnak; olyanokra, akik inkább a



vi-t szeretik; és mindenki másra. Van, akik azt mondják, hogy az Emacs egy jó operációs rendszer, bár elismerik, hogy a Unixban több program van. Bonyolultságából következik, hogy hosszú időbe telik, amíg otthonosan mozoghatunk benne, és inkább azok számára ajánlom, akik naponta fogják használni, hiszen elég pár hetes kihagyás, és máris jó néhányat elfelejthetünk a számtalan billentyűkombinációból.

Az Emacs név a **Richard M. Stallman** által a TECO szerkesztőhöz írt „editing macros” (szerkesztőmakrók) szavak betűiből alkotott mozaikszó. Az Emacs történetét elolvashatjuk a

☞ <http://www.gnu.org/philosophy/stallman-kth.html> (a fájl másolata megtalálható a 24. CD Magazin/Emacs könyvtárban). A vi-hoz hasonlóan az Emacsnak (4. kép) is több változata létezik, például a **Xemacs** (5. kép), ami egy eszköztárral is rendelkezik. Az utóbbit korábban **Lucid** (azaz világos, érthető) **Emacs**-nak nevezték.

Mindkét Emacs-fajta elfogadott, és gyakran együtt hivatkoznak rájuk az Emacs-on szóval. Ahogy az Emacs bejelentkező ablakában olvashatjuk, a GNU Emacs az egyik alkotóeleme a Linux-alapú GNU-rendszereknek, és mint ilyet egyetlen Linux-terjesztésből sem illik kihagyni. Akik GNU-programokat akarnak írni, azoknak otthonosan kell mozogniuk az Emacsban, mint ahogyan a rendszergazdáknak is ismerniük kell a vi-t, ami kötelezően ott van minden Unix-típusú rendszerben, és amit a legvadabb rendszerösszeomlások idején is el lehet indítani. Mindkét program elengedhetetlen kelléke a Unix-kultúrának, a Linux-életérzésnek és a GNU-mozgalomnak.

Ahogy a képeken is látjuk, az Emacs egyszerre több fájl is meg tud nyitni – akár ugyanazon keretben több részre osztva fel a munkaterületet. Számos keretet vagy ablakot tarthatunk nyitva egyidejűleg, futtatható állományokat és képeket nézhetünk meg vele. Ha az Emacs nem támogatott képformátumot talál, akkor külső képnézőt (például ImageMagick) hív segítségül. Programozás közben intézhetjük levelezésünket, honlapokat látogathatunk meg és tölthetünk le akár PostScript formátumban is anélkül, hogy kilépnénk az Emacsból. Amikor elfáradunk, néhány beépített játékkal játszhatunk.

Az Emacsot nem a manapság igen divatos C vagy C++ nyelven írták és írják, hanem a Lisp nyelvet használják erre a feladatra, annak is egy különleges fajtáját, az Elispet, más néven Emacs Lispet. Az Elisp teljes értékű programozási környezet, amivel szöveget és fájlokat kezelhetünk, hálózati alkalmazásokat vagy új felhasználói felületeket építhetünk fel az Emacsban belüli használatra. Az érdeklődők kipróbálhatják még a Jonathan Sajt Emacs Változata (Jonathan's Own Version of Emacs) programot vagy annak menüsített, XJove nevű kiadását, ha beírják a jove vagy xjove parancsokat. Az uemacs vagy más néven MicroEmacs kis teljesítményű gépekre készült, és a 4.0-s változathoz maga **Linus Torvalds** írt javításokat. Utóbbit az em parancssal indíthatjuk.

Az Emacs logikája

Talán igazságtalanul bántam az Emacs-csal, amikor azt mondtam róla, hogy nehézkes és bonyolult, hiszen már rövid használat után beláthatjuk, hogy a fejlesztők egységes és igen egyszerű elvek alapján építették fel a programot. Ezeket az elveket így foglalhatnánk össze: Ha a program használata közben új

szempontok merülnek fel, amelyek megkönnyíthetik a mindennapi munkát, akkor meg kell valósítani azokat. Az új feladatok ellátásához egy vagy több Elisp-függvényt kell megírni, és ezeket a függvényeket a felhasználók számára hozzáférhetővé kell tenni, azaz meg kell adni számukra a lehetőséget, hogy programozás nélkül meghívhassák őket.

Mivel Richard Stallman és társai Unix-környezetben velkedtek, vérükké vált az a szemlélet, hogy egy program csak egy dolgot csináljon, de azt felettébb jól. Követelmény volt a Unixban az is, hogy ezek a programok tetszés szerint összefűzhetők legyenek. S valóban, az Emacsban ott van a mini átmeneti tár, ami lényegében egy miniatűr héj, ahová parancsokat gépelhetünk be. A mini átmeneti tárhoz hasonló parancssori beviteli eszközök eltűntek a mai szövegszerkesztőkből, ha egyáltalán voltak bennük, de a táblázatkezelőkben a szerkesztőlecek még most is megtalálhatók, annak ellenére, hogy az adatokat közvetlenül a cellákba is beírhatnánk. A Unix-követelmények szerint az Emacs-parancsok szintén kötegelhetők, és rendszerint csak egy dolgot tesznek, de azt felettébb jól, hiszen a közel két évtizedes fejlesztés következtében az Emacs igen jól átgondolt és hibamentes lett. Azt is tudjuk, hogy a programozó nem szeret sokat gépelni, annak ellenére, hogy a billentyűzetet mesteri módon tudja kezelni. Magától adódott tehát a felismerés, hogy az egyre szaporodó parancsokat billentyűkhöz és billentyűkombinációkhoz kössék. Tették ezt annál is inkább, mert majd húsz évvel ezelőtt még nem voltak ismertek a grafikus képernyők és a mutatóeszközök. A végeredmény az lett, hogy mára igen nagy számú Emacs-parancs és -változó áll a felhasználók rendelkezésére, akik így igen sok feladatot tudnak egyszerűen és hatékonyan megoldani. Cserébe viszont a kezdőknek viszonylag hosszú ideig kell ismerkedniük az Emacs-környezettel, és nemcsak a számtalan parancs nevét kell megtanulniuk, de a hozzájuk tartozó billentyűkombinációkat is készságszinten el kell sajátítaniuk.

Gondoljunk csak végig, hogy másképp van-e ez a többi, felhasználóbarátan tartott alkalmazásban! Vajon a grafikus felületek kitalálói jobb receptet kínálnak a fenti dilemma orvoslására? Hatásos megoldás lehet, ha csökkentjük a felkínált parancsok számát – áttekinthetőbbé és könnyebben megtanulhatóvá téve a programot. Az ilyen lebutított változatok kielégítőek lehetnek az átlagfelhasz-