

Az Emacs (5. rész)

Ha valamit elrontottunk, azt vissza is állíthatjuk.



A mennyiben valamit eltévesztünk, a *Menu > Undo more* menüpont kiválasztásával vagy az *Undo* ikonnal visszaléphetünk (1. f. kép). A *C-x u* vagy *C-_* ugyanezt teszi. Ha ezeket a parancsokat ismételten kiadjuk, egyre régebbi állapotokat állíthatunk vissza. Ha a visszalépések sorát egy akármilyen jelentéktelen parancs kiadásával (például a kurzormozgató *C-f* megnyomásával) bárhol megtörjük, az addigi visszaállítások is visszavonhatóvá válnak (redo). Az XEmacs a visszaállítás céljára alapértelmezetten általában 20 000 karaktert tud megjegyezni. Ez az érték az *Options/Customize/Emacs/Editing/Undo/Threshold* menüpontban állítható.

A regiszterek

Ha egy szövegrészt többször egymás után be akarunk illeszteni, célszerű a *C-x r s r* teljes billentyűvel egy regiszterbe átmásolni. A regiszterek olyan helyek, ahová szöveget, könyvjelzőket, számokat és állományneveket menthetünk későbbi használatra. A regiszterbe mentettek mindaddig ott maradnak, amíg újabb dolgot nem helyezünk el ugyanoda. A regiszterek neveiként egyetlen betűt használhatunk, azaz a fenti parancsban az utolsó *r* betű a regiszter neve, amit más betűvel is helyettesíthetünk. Később a *C-x r i r* parancssal a szöveget máshová is beilleszthetjük. Ha elfelejtjük, hogy mi található egy adott nevű regiszterben, tartalmát az *M-x view-register RET r* parancssal nézhetjük meg. Könyvjelzőket, azaz olyan szöveghelyeket, ahová később vissza akarunk ugrani, a *C-x r Szóköz d* parancssal menthetünk regiszterbe, aminek a mostani példában *d* a neve. Az ugrást a *C-x r j d* parancs végzi. Lehetőség van még az ablakok vagy keretek helyzetének regiszterekben való mentésére is a *C-x r w v* és az *M-x frame-configuration-to-register RET v* parancsokkal. A visszaállításhoz ismét a *C-x r j v* teljes billentyűt kell használnunk. Most *v* a regiszter neve. Az ablakok és keretek visszaállításakor a beállításban nem szereplők láthatatlanná válnak, de nem törlődnek. Számokat is tehetünk a

regiszterekbe a *C-u szám C-x r n n* parancssal, és az értéküket a *C-u szám C-x r + n* beírásával meg is növelhetjük. Az első példában *szám* számot teszünk az *n* regiszterbe, a másodikban az *n* regiszterben lévő számot növeljük meg *szám* értékkel. Az így mentett számot a *C-x r g n* teljes billentyűvel szűrhatjuk be az *n* nevű regiszterből. Fájlokat is tehetünk regiszterekbe. Ha például a *~/emacs* beállítási fájlba beírjuk a

```
(set-register ?f '(file . "~/emacs"))
```

sort, később a *C-x r j f* parancssal egy lépésben meglátogathatjuk a gyakran használt állományt. Az idézőjelek közé a választott fájl nevét kell írni. A *?f* a regiszter neve.

Átfordító utasítások

A *C-t* átfordító utasítással karaktereket cserélhetünk meg. Ha a *Linuxvilág* szóban a kurzorral ráállunk az *i* betűre, és megnyomjuk a *C-t* billentyűket, akkor a *Linuxvilág* szót kapjuk, azaz a *v* betű helyébe az *i* került, a *i* helyébe pedig a *v*. Az *M-t* szavakat cserél fel. Ha a *transpose Linuxvilág* szavaknál az első szóra állunk a kurzorral, akkor az *M-t* használata a *Linuxvilág transpose* eredményt adja. A *C-x C-t* sorokat, a *C-M-t* kifejezéseket fordít át.

Betűátalakítások

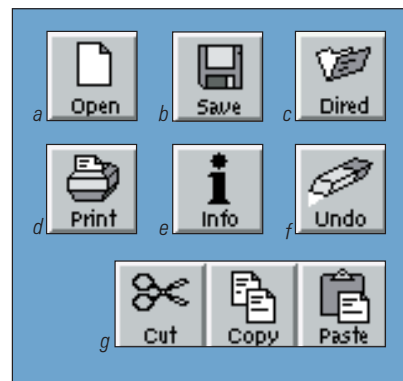
A következő teljes egy vagy több szót billentyűkkel kisbetűssé vagy nagybetűssé alakíthatunk:

M-l – a pont utáni betűket a szó végéig kisbetűssé alakítja

M-u – a pont utáni betűket a szó végéig nagybetűssé alakítja

M-c – (1.) a következő szó első betűjét nagybetűssé és a többi kisbetűssé alakítja, (2.) ha a szó közepén vagyunk, akkor a kurzor alatti betű lesz nagybetűs, a kurzor pedig a szó végére ugrik

A fenti parancsoknak pozitív- vagy negatívszám-kapcsolót is átadhatunk. A *C-u 20 M-u* parancs például 20 egymás után következő szót alakít át nagybetűssé. Mivel a kurzor mindig a szó



1. kép

végére ugrik, a teljes billentyűk nyomogatásával egymás után több szót is átalakíthatunk. Ha közben ki akarunk hagyni egy-egy szót, nyomjuk meg az *M-f* billentyűt.

A következő parancsok nem egy szóra, hanem egy kijelölt területre hatnak:

C-x C-l – a kijelölt területet kisbetűssé alakítja,

C-x C-u – a kijelölt területet nagybetűssé alakítja.

Az *M-c* billentyűnek megfelelő területet átalakító parancs használata ritkább, ezért egyetlen billentyűhöz sem kötötték:

M-x capitalize-region

Ez a parancs a kijelölt terület minden szavának első betűjét nagybetűssé, a többi kisbetűssé alakítja.

Könyvjelzők

A kurzor pillanatnyi helyzetébe a *C-x r m könyvjelzőnév RET* parancssal szűrhatunk be könyvjelzőket, amik a regiszterektől annyiban különböznek, hogy hosszú neveik lehetnek, és rejtve a szövegben az Emacs bezárása után is megmaradnak, mivel a program kimentti őket a *~/emacs.bmk* nevű fájlba:

```
(("emacs-mark"
 (filename .
  "~/home/ratio/.emacs"))
```

```
(front-context-string .
  ↪ "XEmacs to avoid ")
(rear-context-string .
  ↪ "macs-custom for ")
(position . 3371)))
```

A fenti példa az emacs-mark nevű könyvjelzőt elhelyezi az .emacs beállítófájlba a 3371. pont helyzetébe. A már meglévő könyvjelzőket a *C-x r l* teljes billentyűvel listázhatjuk ki, ahol egyúttal szerkeszthetjük is őket. A könyvjelző mód szerkesztési parancsait ilyenkor a *C-h m* súgó parancssal tekinthetjük meg. A *C-x r b könyvjelzőnév RET* parancssal ugorhatunk a kívánt helyre, és beírskor a *NYÍL* billentyűket a mini átmeneti tárlan használhatjuk a már beírt könyvjelzők előhívására. Egy már meglévő könyvjelzőt az *M-x bookmark-delete RET könyvjelzőnév RET* parancssal törölhetünk.

Címkék

Programozás közben gyakorta előfordulhat, hogy nem emlékszünk rá pontosan, hogy az adott függvény mit csinál. Ilyenkor célszerű megkeresni és megnézni, hogy mi van benne. Ehhez nyújtanak segítséget a címkék (tags). Az Emacs képes önműködően címkéket rendelni többek közt a C-, C++-, Java-, Lisp-, Pascal- és Perl-kódhoz. Ezt a munkát most az *etags* program végzi el helyettünk, amely a fájl végződéséből és tartalmából felismeri, hogy melyik programozási nyelvről van szó:

etags file1 file2 ...

A címkéket tartalmazó állomány alapértelmezetten a TAGS címketáblázat fájl lesz, amit a program a munkakönyvtárba tesz. Ha programunkhoz új függvényeket adunk, akkor ezt a fájlt időről időre célszerű frissíteni. Egyszerre több címketáblázat fájlunk lehet, de közülük egyszerre csak egy lehet aktív. Az alapértelmezettől eltérő az *M-x visit-tags-table* parancssal választhatunk. Miután címketáblázatunk elkészült, az *M-címke RET* parancssal kereshetünk benne. Ha nem tudjuk, milyen címkék közül választhatunk, a *Tab* megnyomásával listát kapunk. A listából a középső egérrel a megszokott módon választhatunk. A *C-u M-* teljes billentyű megke-re-si a legutóbb megtalált címke következő előfordulását, ha van ilyen. Negatív kapcsolóval visszaugrik az előzőleg megtalált címkehez, tehát ilyenkor a *C-u M-* teljes billentyűt kell gépel-nünk. Az *M-* billentyű a pillanatnyi

tárolóban nézi meg, hogy nincsen-e még egy találat. Az *M-x list-tags RET fájlnev RET* az állományhoz tartozó címkéket listázza ki.

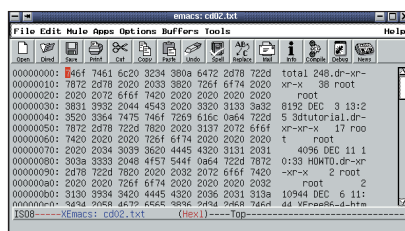
Héjparancsok

Ha héjparancsokat akarunk végrehajtatni, nem kell kilépnünk az Emacsból, elegendő, ha a mini átmeneti tárolóba beírjuk az *M-! parancs RET* parancsot. Például kiíratthatjuk az aktuális munkakönyvtárat:

M-! pwd RET

Ki is listázhatjuk a tartalmát:

M-! ls -l RET



A listázás eredménye egy új Emacs-ablakban jelenik meg. Ha akarunk, az *M-x shell* parancssal egy ablakban új héjat nyithatunk.

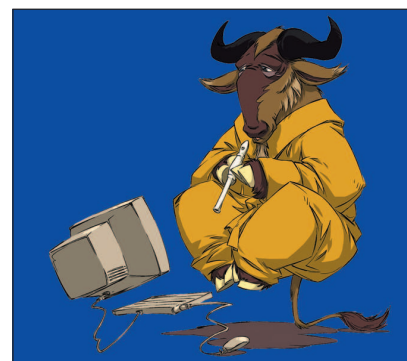
Bináris fájlok szerkesztése

Bináris állományok hexadecimális szerkesztésére a fájlt az *M-x hexl-find-file* parancssal kell behívni, ami működésbe hozza a *Hexl* főmódot, amelyben a következő parancsok használhatók:

C-M-d szám RET – egy tízes számrendszerben megadott bájtot szűr be,
C-M-o szám RET – egy nyolcas számrendszerben megadott bájtot szűr be,
C-M-X szám RET – egy tizenhatos számrendszerben megadott bájtot szűr be,
M-g szám RET – a tizenhatos számrendszerben megadott címre ugrik,
M-j szám RET – a tízes számrendszerben megadott címre ugrik,
C-x I – egy 1 kilobájt nagyságú szöveg-rész elejére ugrik,
C-x J – egy 1 kilobájt nagyságú szöveg-rész végére ugrik,
C-c C-c – kiléphetünk a *Hexl* módból.

Súgó az Emacsban

Az XEmacs rögtön induláskor eligazítást ad arról, miképpen juthatunk segítséghez. Mihelyt megnyomjuk az *F1* gombot, észre fogjuk venni, hogy nem a Windowsban megszokott Súgó rendszerhez jutunk. A kezdő felhasználó nem minden



alap nélkül úgy érezheti magát, hogy ez a Súgó olyanok számára készült, akik már mindent tudnak az Emacsról, és csak az emlékeztetőket akarják frissíteni. A legalul lévő visszhangterületen üzenetet kapunk, hogy gépeljünk be egy ? kérdőjelet. Miután megteesszük, listát kapunk a beüthető billentyűkről.

Elsőként nyomjuk meg az 'a' betűt, majd a mini átmeneti tárolóba írjuk be egy olyan karakterláncot, ami sejtésünk szerint része a keresett parancsnak.

Tegyük fel, olyan parancsokat keresünk, amik tartalmazzák a *language* szót.

Mivel programozók vagyunk, nem szeretünk többet gépelni, mint amennyi feltétlenül szükséges, ezért csak annyit írunk be, hogy *lan*. Miután megkaptuk a keresés eredményét, láthatjuk, hogy az Emacs minden olyan parancsot kilistáz egy ablakban, ami tartalmazza a fenti betűket, mint például a *describe-lang uage-environment* vagy a *balan ce-windows* parancsokat. Gyakorlasképpen nézzük meg, hogy milyen eredményt kapunk a *fon* vagy a *des* szótöredékek begépelésekor! A *q* megnyomásával elhagyhatjuk a *Súgó* ablakot. Most gépeljük be a hasonlóan működő és hasonló feladatot ellátó *M-x apropos* parancsot.

Nyomjuk meg ismét az *F1* billentyűt, majd üssük le a *b* betűt! Most az Emacsban használatos billentyűkombinációkról kapunk listát. A *c* betű megnyomása után azt a billentyűkombinációt kell a mini átmeneti tárla begépelnünk, amiről súgást akarunk kapni. Például a *C-x 5 2* beírása után a következő válasz jelenik meg: *C-x 5 2 runs the command make frame*. Ha ezt a parancsot az *F1 c* előtag nélkül gépeljük be, az Emacs egy új keretet fog létrehozni. Próbáljuk ki ezt is! Az *F1 f Tab* listát készít az Emacs Lisp-függvényekről, az *F1 k* rövid leírást ad a megnyomott billentyű feladatáról, az *F1 w* pedig egy parancs begépelésekor

kinyomatja, hogy milyen billentyűkkel hívhatjuk elő. Az **F1** az utolsó száz begépelt karaktert mutatja meg. Ez jól jön adatvesztéskor vagy hibakereséskor, mivel megnézhetjük, hogy miket is csináltunk. Ha egy menüpont leírását szeretnénk olvasni, írjuk be a **C-h k** billentyűkombinációt, és az egerrel válasszuk ki a kívánt menüt! Próbáljuk ki az **F1** megnyomása után a visszhangterületen látható többi billentyűt is!

A kezdőknek elsőként az oktatófájlban célszerűbb átrágniuk magukat, amit a *C-h t* vagy *FI t* kombinációkkal indíthat el. Megnézhetjük még a man- és infooldalakat a *man (x)emacs* vagy *info (x)emacs* parancsokkal a héjban, vagy elolvashatjuk azokat magában az Emacsban a *Help* menüpont alatt és az XEmacsban az *Info* ikont megnyomva (1 e kép).

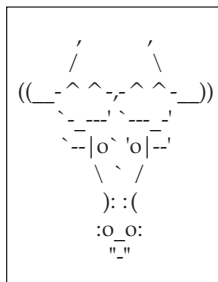
Az *F1 i* szintén az infooldalakat hozza elő, az *F1 F* pedig a *Gyakori Kérdések*-et. A *Súgó*-ban szöközzel lapozhatunk, és a *q* megnyomásával léphetünk ki belőle.

Utószó

Mostani rövid sorozatomhoz a SuSE Linux 7.2 profi kiadását használtam. Feltehető, hogy más terjesztések alapértelmezetten nem tartalmazzák a legelső

részen ismer-
tetett szerkesz-
tők mindegyikét,
ezért pillanat-
képeket készí-
tettem róluk,
amelyeket
a CD-melléklet
Magazin/Emacs
könyvtárába
helyeztem.

Néhány más, a témához kapcsolódó anyag is található a teljesség igénye nélkül. Nem hiszem, hogy az olvasók közül bárki egy ültő helyében elejétől a végéig el fogja olvasni ezt az írást, de nem is ez volt a célom. Az Emacs iránt érdeklődők üljenek le a gép elé, és kezdjék el használni a fent leírtak alapján. Nem garátlhatom, hogy az Emacs mindenkinél ugyanígy fog működni, de nagy eltérések nem lehetnek. A billentyűkombinációkat nem bemagolni kell, hanem készszíntben az ujjainkban rögzíteni. Vannak olyanok, akik már évek óta használják az Emacsot, és nem tudják megmondani, hogy ezt vagy azt a parancsot milyen teljes billentyűvel lehet előhívni, de amikor leülnek a számítógép elé, gondolkodás nélkül beütik azt. Lehet,



hogy őrületségnek tűnik a sok felsorolt billentyű és parancs megtanulása, és még nagyobb őrületségnek az, hogy még azokat is megtanuljuk, amelyeket most nem említettem meg, de számos gyakorlott programozó egybehangzó véleménye szerint az Emacs megismerése hosszú távon kifizetődő. Nyilvánvaló, hogy a *vi* vagy az Emacs más, mint amit más operációs rendszerekben eddig megszoktunk, de az a tény, hogy furcsa és szokatlan, még nem lehet a minősítés alapja. Ha valakinek van olyan angol, német vagy magyar nyelvű leírása az Emacsról, ami nem szerepel a CD-mellékleten, szívesen fogadnám. Érdekes lenne arról is hallani, hogy ki és miképpen alakította az Emacst a saját lelkületének megfelelőre.



Száló István
(ratiosoft@freemail.hu) tanár,
immár több mint másfél
évtizede foglalkozik progra-
mozással, de csak a Java és
a Linux megismerése után
hogy mi is az igazi progra-
bb írása megjelent már a hazai
technikai lapokban.

