

## Bevezetés az Emacs használatába

Az Emacs a mindennel felszerelt szövegszerkesztő. Tanuljuk meg az alapjait, a végén talán még a határidőnaplónkat is ebben fogjuk tartani.

**E** cikk gyorstalpaló Emacs-tanfolyam szeretne lenni. Feltételezzük, hogy az Emacs (ejtsd: imex) már telepítve van és fut (a legtöbb Linux-terjesztésnél ez nem gond). Az Emacs beépített oktatóprogramját is ismernek vesszük. Az Emacs segítségével történő programfejlesztés nem témája ennek a cikknek, mert már írtam róla a Linuxvilág 2002. júniusi számában.

Az Emacs indításához gépeljük be az `emacs &` parancsot. Az `&` jel az Emacsot a háttérbe teszi; s mivel az X jeleníti meg, ez a beállítás megfelelő.

Az Emacs a konzolon is futtatható, ekkor `&` nélkül adjuk ki a parancsot. Ha egy `xterm` ablakban szeretnénk futtatni, új ablak megnyitása nélkül, az `emacs -nw` parancsot kell használnunk. A konzolos és az `xterm`es üzemmód nagyon hasznos, ha nincs X, például amikor egy távoli kiszolgálón SSH-kapcsolaton keresztül dolgozunk. Az SSH beállítható az X továbbítására, és ha ezt engedélyezzük, akkor futtathatjuk a távoli Emacsot (és más X-alkalmazásokat).

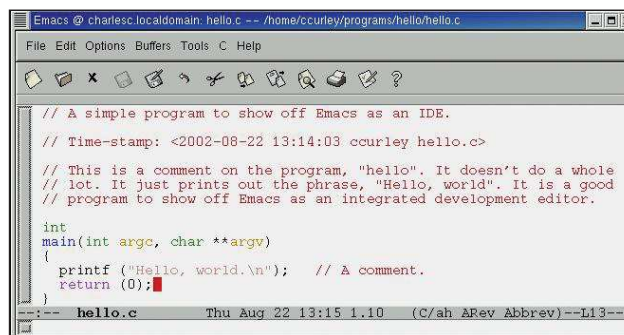
Ha még nem jutottunk el a beépített Emacs-tanfolyam végére, most tegyük meg. Nyomjuk meg a `CTRL-H` billentyűkombinációt, majd a `T` billentyűt. Az oktatóprogram a számítástechnika őskorában született (1985), így nem veszi figyelembe a kurzorvezérlő billentyűket és más korszerű kényelmi szolgáltatásokat. Az Emacs támogatja ezeket, de az oktatóprogram nem veszi figyelembe. Ha nem is tanuljuk meg, nem árt tudunk bizonyos Emacs-gyorsbillentyűk létezéséről, a `bash` és még sok GNU-program is beállítható a használatukra. Például a `CTRL-B` és a `CTRL-N` billentyűkombinációknak pontosan ugyanaz a hatása az Emacsben, mint a `bash`ban. Valójában az Emacs stílusú gyorsbillentyűk az alapértelmezettek a `bash`ben.

Az oktatóprogram segítségével elsajátíthatjuk az alapvető kurzormozgatási ismereteket, az Emacs-parancsok megszakítását, az Emacs ablakkezelését, a szövegtárolók (bufferek) és a fájlok kapcsolatát stb. Talán azt a legszükségesebb megjegyeznünk ebből, hogy a mozgatóbillentyűk általában adott billentyűk (az F előre), és a mozgás tartományát a módosítók határozzák meg. Például a `CTRL-F` a kurzort egy betűhellyel mozgatja előre, a `M-F` viszont egy szóval előbbre helyezi (a `M` az Emacs jelölése a Meta billentyűre, ami a legtöbb billentyűzeten ALT-ként található meg).

Az Emacs jóval a webböngészők megjelenése előtt létezett már, ezért a „keret” kifejezést az X-es ablakra, az „ablak” kifejezést pedig a kereten belüli szakaszra használja. Mivel ez a cikk az Emacsról szól, az Emacs szóhasználatát fogjuk alkalmazni. A `CTRL-X 2` billentyűkombináció hatására egy új ablak jelenik meg, vízszintes elválasztással. A függőleges elválasztás új ablakhoz a `CTRL-X 3` billentyűkombinációt kell alkalmaznunk. A `CTRL-X 5` új keretet nyit, a `CTRL-X 0` és a `CTRL-X 5 0` bezárja a pillanatnyi ablakot, illetve keretet. Az oktatóprogram másik kiemelt része az Emacs növekményes keresőparancsainak ismertetése. Nagyon megkönnyítik az életet, ezért tanuljuk meg jól őket!

Az Emacs súgórendszere a `CTRL-H` billentyűkombinációval

hívható elő. A `CTRL-H?` hatására a súgórendszer különböző részeit felsoroló lista jelenik meg. Az inforendszer (`CTRL-H I`) az FSF infoformátumában elérhető dokumentumok böngészését teszi lehetővé. Ez egy faszervezetbe rendezett hiper-



```
Emacs @ charles.localdomain: hello.c -- /home/ccurley/programs/hello/hello.c
File Edit Options Buffers Tools C Help
// A simple program to show off Emacs as an IDE.
// Time-stamp: <2002-08-22 13:14:03 ccurley hello.c>
// This is a comment on the program, "hello". It doesn't do a whole
// lot. It just prints out the phrase, "Hello, world". It is a good
// program to show off Emacs as an integrated development editor.
int
main(int argc, char **argv)
{
  printf("Hello, world.\n"); // A comment.
  return (0);
}
```

1. kép Az Emacs C-módja a nyelvi elemek színezésével. A behúzásokról a C-mód gondoskodott

szöveges rendszer, még a World Wide Web előtti időkből. Az Emacs szolgáltatásainak Info-fejezeteihez a `CTRL-H CTRL-F` billentyűkombinációval juthatunk el. Ez a fejezet a pillanatnyi fő- és almmódról (részletesen lásd később), az Emacsre vonatkozó felhasználási szerződésről és garanciáról jelenít meg adatokat. Az Emacs súgóját az Emacs jeleníti meg, ezért az oktatóprogramban megtanult kurzormozgató billentyűkombinációk a súgóban is használhatóak.

### Ismerkedés a módokkal

Az Emacs legelső súgóoldala szerint az Emacs bővíthető, testreszabható, önleíró, valós idejű megjelenítéssel dolgozó szövegszerkesztő. Bővíthető, mert Emacs Lisp nyelven, más néven Elisp nyelven írták, ami a Lisp nyelvnek kimondottan az Emacs és a szövegfeldolgozás számára testreszabott változata. A Emacs tudását Elisp nyelvű kód írásával bővíthetjük. Ezenkívül az Emacs viselkedése a meglévő elisp-változók értékeinek megváltoztatásával testreszabható. Az önleíró jelző talán kicsit túlzás, de az Elisp tényleg támogatja a programozót a kódja dokumentálásában. Láttuk már, hogy részletes súgó áll a rendelkezésünkre.

Az Emacsot a felhasználó is testreszabhatja egy adott feladat elvégzésére, ezt az úgynevezett főmód váltásával érjük el. Egy szövegtárolóban egyszerre csak egy főmód lehet működő, de menet közben is lehet főmódot váltani. Például CGI-programok írásánál hasznos lehet a Perl-mód és a HTML Helper-mód közti váltogatás.

A szövegtárolóban pillanatnyilag érvényes mód azonosításához vessünk egy pillantást a módsorra. Zárójelben található egy vagy több mód, és a pillanatnyi főmód az első a sorban. Nem minden almmód azonosítja magát a módzárójelekben, de a tevékenységük magától értetődik, például a *Column Number* (oszlopszám) mód.

## Főmódok

A főmódok általában az állományok kiterjesztéséhez vannak kötve. Az `auto-mode-alist` Lisp-változó gondoskodik a hozzárendelésekről, és meg fogjuk mutatni, hogyan adhatók hozzá további hozzárendelések. Az Emacs a parancsfájlok első sorának különleges bejegyzéséből is képes meghatározni az állomány típusát, például a következőhöz a Perl-módot rendeli:

```
#! /usr/bin/perl
```

A módot mindig kikényszeríthetjük, ha a dokumentum első sorába a kívánt mód nevét `-*-` jelek közé tesszük:

```
# -*- shell-script -*-
```

Kézzel a `M-X` mód neve parancssal válthatunk egyik módról a másikra, például a `M-X perl-mode` az Emacs-t a Perl-programok szerkesztéséhez használható főmódba váltja át.

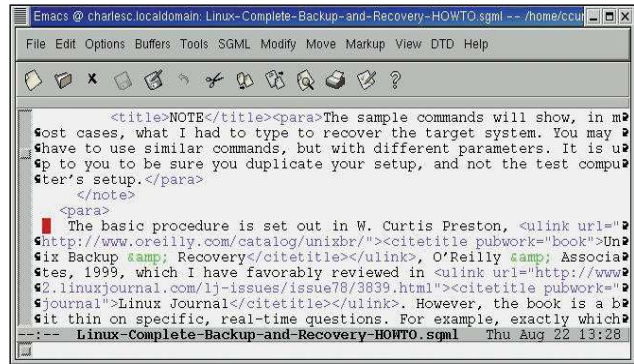
A főmódok számos hasznos szolgáltatást nyújtanak. Általában az adott állománytípusra jellemző behúzási szabályokat és áttekinthető megjelenítést tesznek lehetővé. Sokszor lehetőség nyílik a megjegyzések beszúrására egy rövid billentyű-kombináció segítségével. Egy szövegterület egyszerű mozdulattal helyezhető megjegyzésbe: az `M-X comment-region` parancssal. Ha minden szerkeszteni valónkat az Emacs-szel végezzük, az többek között azért hasznos, mert az egyik főmódban megismert szolgáltatás (valamint a gyorsbillentyűje és menüparancsa) nagy valószínűséggel megvan a többiben is, azaz ha képesek vagyunk a C-fájlokat szerkeszteni az Emacsben, akkor valószínűleg az SQL-fájlok szerkesztése is menni fog (1. kép).

A főmódok általában színekkel emelik ki a nyelvi elemeket. A nyelvi elemek és a színek egymáshoz rendelése önműködően történik. Például a megjegyzések pirosak, az adattípusok zöldek és a karakterláncok világos pirosak. Az Emacsben történő szerkesztésnek az a másik előnye, hogy a színek hozzárendelése minden módnál ugyanaz, azaz a megjegyzések pirosak – akár assembly kódot, akár XML-t szerkesztünk. A főmódok módosítják a billentyűk működését, általában a `TAB` és a `DELETE` billentyűt. A főmódok módfüggő parancsait a `CTRL-C` előtag vezeti be. Például a `PSGML`-módban a `CTRL-C CTRL-V` parancssal lehet az `SGML`-dokumentumot érvényesíteni.

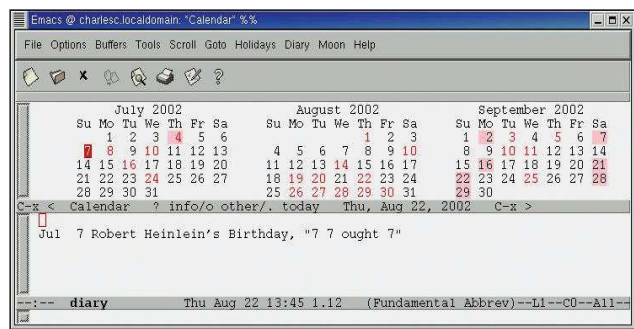
Az egyik legnagyobb tudású főmód **Lennart Staflin** `PSGML`-módja (lásd a *Kapcsolódó címeket*). Lehetővé teszi `SGML`- és `XML`-elemek beszúrását, és önműködően a `C`-móddal összemérhető áttekinthető megjelenést ad a dokumentumnak.

A `PSGML` is színezi a nyelvi elemeket és egyéb jó tulajdonságai is vannak, de ezeken kívül beolvassa a `DTD`-t is, és kikényszeríti az elemek helyes egymásba ágyazását. Például a `DocBook`-ban nem engedi meg egy `<sect4>`-nak közvetlenül egy `<sect1>` környezetbe történő beszúrását. A `PSGML` egyúttal az előtérprogram az érvényesítőhöz (2. kép).

A következő főmódok lehetnek még hasznosak szinte bárki számára: `Dired`-mód, `Ediff`-mód, `W3`, a `Calendar` (naptár) és `Diary` (határidőnapló). A `Dired`-mód a könyvtárak szerkesztésére való. Könyvtárról könyvtárra ugrálhatunk, rámehetünk fájlokra és szerkeszthetjük a fájlok metaadatait, például a hozzáférési jogokat és a tulajdonost. A `Dired`-mód egyik legjobb tulajdonsága, hogy képes több fájlban keresni, és megjelölni a találatokat. A találatokra sorban rámehetünk és szerkeszthetjük őket. Így lehetséges a fájlok „tömeges” kezelése, például az átnevezésük vagy a törlésük.



2. kép A Linux Documentation Project egy dokumentumának szerkesztése a DocBook SGML DTD-vel Emacsben. Az elemek és az entitások színekkel vannak kiemelve



3. kép Az Emacs naptára és határidőnaplója

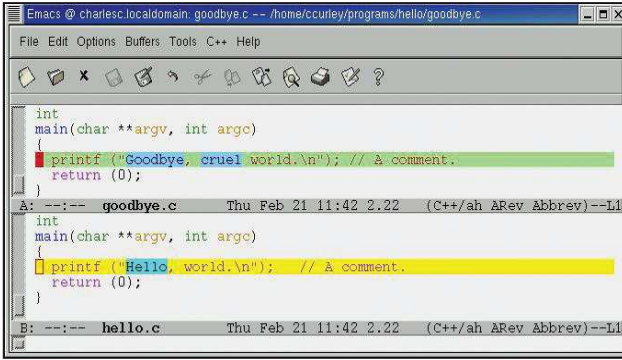
A naptár, illetve a határidőnapló meglehetősen hasznos mód. Az átlagos naptárprogramoktól eltérően a `Calendar`-mód lehetővé teszi a dátumok kezelését és átváltását a Gergely-naptár, a Juliánus-naptár, a kopt, a héber és az iszlám naptár között. Ha valami különlegességre vágyunk, a legközelebbi feljegyzésünket dátumozzuk a perzsa vagy a maja naptár szerint. Esetleg a következő hibajelentésünket a francia forradalmi naptár szerint küldjük el a Free Software Foundationnek (3. kép). A jelentéktelen naptáraknál hasznosabb a határidőnapló, amiben találkozókat, évforduló-emlékeztetőket, ismétlődő eseményeket (például „minden hónap harmadik csütörtöke”) és más eseménytípusokat jegyezhetünk be. Ha megadjuk az esemény idejét, az Emacs figyelmeztet az idő közeledtekor. A határidőnapló nem pusztán önmagában hasznos, hanem mivel Emacsben fut, minden olyan számítógépen működik, amire létezik Emacs – márpedig a legtöbb számítógépre van. A naplóállomány szintén hordozható.

Az `Ediff`-mód foltok kiválasztott alkalmazására használható. Arra is jó, hogy több számítógépen frissítsük az állományokat, például a `.emacs`- és a naplóállományokat. Mivel kiválasztható, az `Ediff` lehetővé teszi a változtatások mindkét irányú átvitelét. Ez fontos lehet, ha találkozókat állítunk be hordozható gépünkön, miközben a titkárnő szintén találkozókat állít be az asztali gépünkön (4. kép).

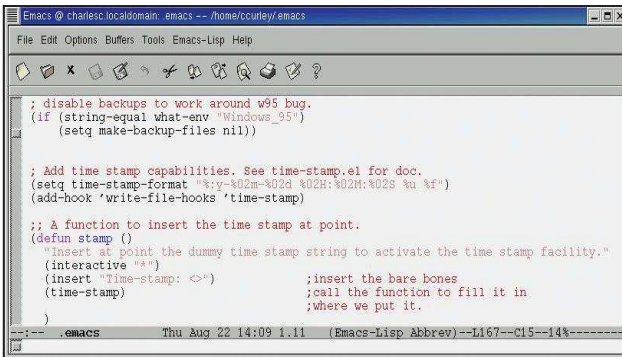
Ha a világhálót szeretnénk böngészni, vizsgáljuk meg **William M. Perry** `W3`-módját – ez egy Emacs Lisp nyelven írott webböngésző.

## Almódok

Az almódok, azaz a bővítmények a főmódokat egészítik ki. A legtöbb almód a főmódtól függetlenül működik, így külön-



4. kép Két fájl összehasonlítása. Az Emacs nemcsak a megváltozott sorokat mutatja, hanem a sorokon belüli különbségeket is



5. kép A szerző .emacs állományának szerkesztése az Emacsben, példa az Emacs Lispre



6. kép Az Emacs karakteres felületen indítása

féle dokumentumokon használható. Például a Show Paren mód a zárójeleket párosítja össze. Ez a C-programozók számára is hasznos, még Text-módban is, de Lisp-programozók számára nélkülözhetetlen segédeszköz.

Az almodokat tetszés szerint ki- és bekapcsolhatjuk. Programozásnál például az Auto Fill-mód (szövegkitöltés, sortörés, bekezdések) hasznos a megjegyzések írásánál, de idegesítő a kód írásánál.

Bizonyos almodok mindenre kiterjedők, vagyis az összes szövegtárolóra vonatkoznak, mások csak az adott szövegtárolón belül érvényesek. Egy adott mód bekapcsolásához a neve után a -mode karakterláncot írjuk a parancssorba. Például a Parentheses-mód indításához nyomjuk le a M-X billentyűkombinációt, majd írjuk be a show-paren-mode parancsot. A mód kikapcsolásához újra ezt a parancsot futtassuk.

Néhány hasznos szövegtárolón belüli almod: Abbrev-mód (menet közbeni önműködő javítás), Auto Save-mód, Font-Lock-mód (színes szövegkiemelés), Flyspell mód (menet közbeni helyesírás-ellenőrzés) és Overwrite-mód. Két hasznos almod, ami minden szövegtárolóra érvényes, a Line Number és a Column Number mód. Ezek a kurzor pillanatnyi helyzetét írják ki a módsorba, általában a jobb oldalra.

Egy további hasznos mód az Ispell, amivel a szövegtároló helyesírás-ellenőrzését végezhetjük el. Külön almodokkal rendelkezik elektronikus levelek, programszövegekbe ágyazott megjegyzések és karakterláncok, illetve más különleges alkalmazási területek ellenőrzésére.

### A .emacs fájl

Az Emacs testreszabásának kulcsa a ~/.emacs beállítóállomány. A rendszergazdák általában egy egész rendszerre érvényes beállítóállományt készítenek. Ha ez nekünk nem felel meg, a saját beállítóállományunkban megkérhetjük az Emacsot, hogy ne vegye figyelembe. Hibakereséskor hasznos lehet, ha az emacs -q parancsot kiadva az Emacsot a beállítóállomány elolvasása nélkül indítjuk el. A beállítóállomány nem más, mint egy kis Elisp-program, ami tetszésünk (vagy a rendszergazda tetszése) szerint állítja be az Emacsot (5. kép).

A beállítóállományban változóknak is adhatunk értéket. Én az alábbi módon állítottam be a HTML Helper-mód egyes változóit:

```
(setq html-helper-do-write-file-hooks t)
(setq html-helper-build-new-buffer t)
(setq html-helper-address-string "<a href=\
"mailto:ccurley@trib.com\">Charles Curley</a>")
```

Rövid, hasznos függvények vagy makrók is bekerülhetnek a .emacs állományba, például a következő makró a mai dátumot szűri be:

```
(defun insert-date ()
  "A mai dátum beszúrása az \"insert-date-
  ↪format\" változó szerint."
  (interactive "*")
  (insert (format-time-string insert-date-
  ↪format
  (current-time))))
```

A makrókhoz gyorsbillentyűket vagy billentyűkombinációkat is rendelhetünk. Ezek után a billentyűkombinációval is elindítható a makró; például a dátumbeszűrő makró megírása után az F3 billentyűt a következő sorral rendelhetjük hozzá:

```
(global-set-key [f3] `insert-date)
```

Ez a módszer arra is alkalmas, hogy megváltoztassuk a billentyűzetkiosztást. Ha nem tetszenek az Emacs bizonyos hosszú billentyűkombinációi, megváltoztathatjuk őket. Az Emacs testreszabásának másik módja a *Customize* menü használata, amit a M-X customize parancs vagy az *Options* legördülő menü segítségével hívhatunk elő. A kiterjedt menürendszerben a felhasználók megváltoztathatják a változók értékét, és a változásokat menthetik a saját beállítóállományukba.

## Az Emacs mint kiszolgáló

Számos program, például a mutt és a crontab külső programot hív meg szövegszerkesztőként. Az Emacs használatát úgy engedélyezhetjük, hogy az Emacs-t kiszolgálóként indítjuk. Írjuk be a következő sort a `.emacs` állományba:

```
(server-start)
```

Ezt követően állítsuk be az EDITOR vagy a VISUAL környezeti változót az emacsclient értékre. Bash használata esetén a következő sort adjuk hozzá a `/etc/bashrc` állományhoz vagy a saját `~/bash_profile` állományunkhoz:

```
export VISUAL=emacsclient
```

Ezek után a crontab -e végrehajtásakor vagy üzenet szerkesztésekor a mutt-ban a meglévő Emacs-munkamenetben

## KAPCSOLÓDÓ CÍMEK

Az Emacs beépített súgórendszere CTRL-H  
 A `/etc` könyvtár az Emacs-fában  
 A GNU Emacs honlapja  
 ➔ <http://www.fsf.org/software/emacs/emacs.html>  
 A szerző `.emacs` állománya  
 ➔ <http://w3.trib.com/~ccurley/emacs.init.html>  
 Az Emacs-webgyűű  
 ➔ <http://www.gnusoftware.com/WebxRing>  
 Az „emacs for vi Users” ➔ <http://grok2.tripod.com>  
 A „GNU Emacs Tutorial” Sarir Khamsi-tól (régí, de hasznos)  
 ➔ <http://www.futureone.com/~sponge/tutorial/emacs/index.html>  
 A „very unofficial dotemacs home” Ingo Koch-tól  
 ➔ <http://www.dotemacs.de>  
 A „The Emacs Beginner's HOWTO” Jeremy D. Zawodny-tól ➔ <http://www.tldp.org/HOWTO/Emacs-Beginner-HOWTO.html>  
 A „How do I make common modifications to my Gnu Emacs .emacs file? GNU Emacs Manual – Init Examples”, rupe ➔ <http://www.yak.net/fqa/124.html>  
 Emacs-típek: „Nifty ways to get your work done in /the/ editor” ➔ <http://www.portico.org/index.php3?catList=28>

### A cikkben ismertetett módok, amelyek nem részei az Emacsnek

HTML Helper-mód  
 ➔ <http://www.gest.unipd.it/~saint/hth.html>  
 Java Development Environment for Emacs (JDEE)  
 ➔ <http://jdee.sunsite.dk>  
 PSGML-mód  
 ➔ [http://www.lysator.liu.se/projects/about\\_psgml.html](http://www.lysator.liu.se/projects/about_psgml.html)  
 LDP Author Guide, Mark F. Komarinski, Jorge Godoy és David C. Merrill munkája ➔ <http://www.tldp.org/LDP/LDP-Author-Guide>. Az SGML és az XML telepítésében és használatában segít, különösen a DocBookéban, továbbá az Emacs testreszabását is ismerteti.  
 Emacs/W3 v4.0 ➔ <http://www.cs.indiana.edu/elisp/w3>

szerkeszthetünk, ahelyett, hogy egy új Emacs-példány indulására kellene várnunk. A szerkesztés befejezésekor a CTRL-C K helyett a CTRL-C # paranccsal fejezhetjük be a szövegtároló szerkesztését, és léphetünk ki az emacsclient programból. Az emacsclient működéséhez az Emacs programnak – amikor a külső program meghívja – már futnia kell. Ez összhangban van az Emacs használatának javasolt módjával, ami szerint bejelentkezéskor indítsuk el az Emacs egy példányát, és amíg ki nem jelentkezünk, hagyjuk futni. Az emacsclient használatának egyik eredménye az, hogy egyszerre csak egy Emacs-példány fut. Bár a memória napjainkban olcsó, ez nem volt mindig így. Még ma is, ha a Linuxot hordozható számítógépen vagy régi számítógépen futtatjuk, a takarékos memória-felhasználás jó dolog.

Érdeemes beállítani, hogy az Emacs Mail-módban szerkessze a leveleket. Ha mutt-tal használunk, a következő sorokat adjuk hozzá a `.emacs` állományhoz:

```
;; Ha a fájlnev /tmp/mutt kezdetű, lépünk
;; mail-mode üzemmódbba
(setq auto-mode-alist (append (list (cons
  ↪ "^/tmp/mutt" 'mail-mode))
  auto-mode-alist))
```

Természetesen szeretnénk megfelelni az RFC-kben leírt netikettnek is, ezért engedélyezzük az Auto Fill-módot a levél szerkesztésekor. A legtöbb főmód olyan, hogy a módba történő belépéskor és módból való kilépéskor végrehajthatunk egy parancsot. A következőképpen utasíthatjuk a Mail-módot arra, hogy belépjen az Auto Fill-módba:

```
(defun my-mail-mode-hook ()
  (auto-fill-mode 1)
  )
(add-hook 'mail-mode-hook
  'my-mail-mode-hook)
```

A levél megírása után, ha idegesíteni szeretnénk az NSA-t, használjuk a Spookot. Ha tiltakozni szeretnénk a Communications Decency Act ellen (ez helyes cselekedet), és jó pár amerikai politikust szeretnénk idegesíteni, akkor nézzük meg a Bruce-t.

Végül mielőtt befejeznénk ismerkedésünket e vad és buja szövegszerkesztővel, hadd hívjam fel a figyelmet a `/etc` könyvtárra (az Emacs könyvtárfában). Ez a könyvtár sok hasznos dokumentumot tartalmaz, például az Emacs angol nyelvű referenciakártyáját forrásban (`refcard.tex`) és PostScriptben (`refcard.ps`). A referenciakártya különböző nyelvű fordításai is megtalálhatók itt. Van még itt némi háttéranyag az Emacsról és a GPL egy példánya.

Van egy dolog, amit ritkán találunk meg (legfeljebb véletlenül) a szabadalmaztatott programokban, de az Emacsben jelen van: a humor. Nézzük meg a hibajelentést a 2199-es évből, a Spook-mód szövegtárolóját, az Emacs rövidítés néhány magyarázatát stb. Végül, ha igazán szeretnénk megdolgoztatni a betűkészlet-kiszolgálónkat, nyissuk meg a HELLO állományt.

*Linux Journal 2003. március, 107. szám*

### Charles Curley

(w3.trib.com/~ccurley) Linux-oktató és szakíró.  
 Természetesen ezt a cikket is Emacs-szel írta, Linuxon.