

Az Emacs (4. rész)

Hogyan kezeljük az Emacsban az ablakokat és a kereteket?

Elsőként a munkaterületet függőlegesen osszuk fel két részre a **C-x 2** paranccsal, úgy, hogy az egyik ablak a másik fölött legyen, majd a **C-x 1** paranccsal állítsuk vissza az eredeti állapotot! Most üssük be a **C-x 3** billentyűkombinációt, amivel vízszintesen osztjuk ketté a munkaterületet úgy, hogy az ablakok egymás mellé kerüljenek! Folytassuk a felosztásokat tetszőlegesen sorrendben, amíg jó sok vízszintesen és függőlegesen felosztott ablakunk nem lesz, majd a **C-x 0** paranccsal zárjuk be őket egyenként! Figyeljük meg, hogy a **C-x 1** a pillanatnyi kivételével minden ablakot bezár! A sok kinyitott ablak közt egérgattintással vagy a **C-x o** billentyűkkel válthatunk. A kettéosztott ablakokban a pillanatnyi átmeneti tároló tartalma fog megjelenni, ha új tartalommal akarjuk megtölteni, akkor új fájl kell megnyitnunk, vagy a meglévő átmeneti tárolók közül kell választanunk.

Az ablakok az egérrel átméretezhetőek, de a következő billentyűkombinációk is használhatók erre a célra:

M-x shrink-window – úgy csökkenti az ablak méretét, hogy az alsó keretet felfelé viszi,
C-x ^ – úgy növeli az ablak méretét, hogy az alsó keretet lefelé viszi,
C-x { – az ablakot keskenyíti,
C-x } – az ablakot szélesebbé teszi.

A **C-x +** parancs minden ablakot azonos magasságúvá tesz, a **C-x -** pedig csökkenti az olyan ablak magasságát, amelyekben sok üres sor van alul. A felszabaduló helyet a többi ablak kapja. Ha két ablakot nyitunk meg egymás mellett, gyakran előfordulhat, hogy az egyikbe gépelünk, a másiktól pedig olvasunk valamit. Ilyenkor nagyon jól jön, hogy a **C-M-v** paranccsal a másik ablakban előre és lapozhatunk.

Talán lehet valami logikát találni az ablakokhoz és a keretekhez kapcsolódó teljes billentyűk számozásában.

Tekintsük csak át őket még egyszer!

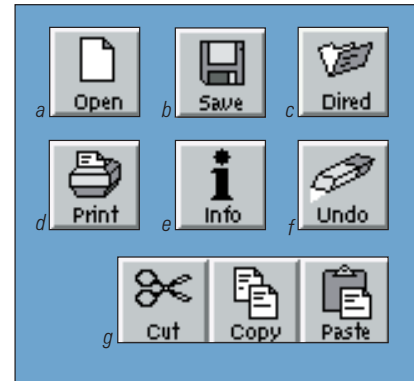
C-x 0 – törli a pillanatnyi ablakot.
C-x 1 – a pillanatnyi ablak kivételével minden ablakot töröl.
C-x 2 – az ablakot függőlegesen osztja ketté.

C-x 3 – az ablakot vízszintesen osztja ketté.
C-x 4 0 – törli az ablakot és a hozzá tartozó átmeneti tárat is.
C-x 4 b – ugrás a másik ablakban lévő átmeneti tárhoz.
C-x 4 d – egy másik ablakban nyitja meg a *Könyvtár Szerkesztőt*.
C-x 4 f – új ablakban nyit meg egy fájlt.
C-x 4 r – új ablakban nyit meg egy fájlt – csak olvasásra.
C-x 4 C-o – egy másik ablakban mutatja meg az átmeneti tárat.
C-x 4 m – egy új ablakban levelet írhatunk.
C-x 4 . – egy címkét egy másik ablakban keres meg.
C-x 5 0 – bezárja a pillanatnyi keretet.
C-x 5 2 – ugyanazt az átmeneti tárat új keretben nyitja meg.
C-x 5 b – az általunk választott átmeneti tárat új keretben nyitja meg.
C-x 5 f – a fájlt új keretbe hívja be.
C-x 5 r – új keretben csak olvasásra nyit meg egy fájlt.
C-x 5 d – új keretben nyitja meg a *Könyvtár Szerkesztőt*.
C-x 5 o – kiválasztja a következő keretet.
C-x 5 . – másik keretben keres meg egy címkét.
C-x 5 m – egy új keretben levelet írhatunk.
C-x m – levelet írunk a pillanatnyi ablakban.

A **C-z** az **X** nélküli parancsoron felfüggeszti az Emacs működését, az **XEmacs**ot pedig ikonméretűvé csukja össze. Ha a főalkalmazás mellett más keretek is nyitva vannak, a **C-z** csak a működő keretet zsugorítja össze.

Kurzormozgatás

Ha a betöltött fájlunk elég hosszú, örömmel tapasztalhatjuk, hogy az ismerős **PAGE UP** és **PAGE DOWN** billentyűkkel lapozhatunk a szövegben, és a le, fel, jobbra és balra mozgó kurzorbillentyűk a megszokott módon működnek. A bal egérgombbal is lapozhatunk, ha többször rákattintunk az **XEmacs**-ablakban található pörgetőrudakra, de ha a középső egérgombbal kattintunk



1. kép



2. kép

ugyanoda, a szöveg azonnal a kattintás által kijelölt szöveghelyre ugrik. A **HOME** a sor elejére, az **END** a sor végére ugratja a kurzort, míg a **CTRL+HOME** a fájl elejére, a **CTRL+END** pedig a fájl végére ugrik. A **DELETE** és az **INSERT** is a várakozásoknak megfelelően viselkedik: a **DELETE** egy karaktert töröl, az **INSERT** pedig a felülírási módot váltogatja a beszúrási móddal. Az Emacs természetéből fakad, hogy ugyanezeket a feladatokat billentyűkombinációkkal is meg lehet ismételni. Például a **C-v** előre, az **M-v** visszafelé lapoz egy képernyőnyit, a **C-x-<** balra, a **C-x->** pedig jobbra görget. Érdekes hatása van a **C-l** billentyűkombinációnak. Vigyük a kurzort valamelyik, a képernyő szélén lévő sorba, jegyezzük meg az ott lévő szavakat, majd nyomjuk meg a **C-l** teljes billentyűt. A parancs hatására minden újrarajzolódik, de a kurzort tartalmazó sor a képernyő közepére fog kerülni. Ha ezzel végeztünk, próbáljuk ki a le, fel, jobbra, balra nyilakat úgy, hogy közben a **CTRL** vagy az **ALT** billentyűket is nyomva tartjuk.

Ahogy korábban említettem, az **ESC** helyettesítheti az **Meta** billentyűt, próbáljuk ki tehát az **Esc-v** kombinációt is! Nálam működött. A le, fel, jobbra és balra mozgatás elérhető a **C-n** (next line azaz következő sor), a **C-p** (previous

line, azaz előző sor), *C-f* (forward, azaz előre) és a *C-b* (backward, azaz visszafelé) lenyomásával. Az előre és a hátra a *Meta* billentyűvel is kombinálható (*M-f* és *M-b*), ilyenkor a kurzor nem egy karaktert mozdul el, hanem egy egész szót ugrik előre vagy hátra. Figyeljük meg, hogy hátrafelé ugráskor a szó első betűjére ugrik, előreugráskor viszont a szó utáni első üres karakter helyére! *C-a* a sor elejére, *C-e* a sor végére ugrik. Hasonló viselkedést nyerünk, ha az *M-a* és *M-e* kombinációt alkalmazzuk, de most a kurzor a mondat elejére vagy végére visz bennünket. Programozás közben az Emacs felismeri, hogy egy kifejezésnek vagy függvénynek hol van az eleje vagy a vége. A *C-M-a* és *C-M-e* kombinációk ilyenkor egy függvény elejére vagy végére ugranak. A *C-M-b* egy kifejezés elejére, a *C-M-f* pedig a végére visz. *M-<* a fájl elejére, *M->* a fájl végére ugrik. Megjegyzem, a *<>* zárójelek nemcsak a magyar billentyűzeten nehezen hozzáférhetőek, ahol az ALT GR-t kell megnyomni, de az angolon is csak a SHIFT megnyomásával hozhatók elő. Ez valószínűleg az ALT és az ALT GR vagy az ALT és a SHIFT egyidejű megnyomását jelenti. Nálam az *Alt-Alt Gr <*-változat ritkán hozta a várt eredményt, helyette inkább az *Esc-Alt Gr <* kombinációt használom, ha használom. Láthatjuk, hogy a táblázat közepén a pont szó ül. Ez ismét olyan Emacs-sajátosságra utal, ami még a TECO nyelv idejére megy vissza, ahol a "." eredetileg egy parancs volt, ami megadta a pont helyén lévő értéket. A pont mindig a kurzor alatti karakter elé mutat (2. kép).

Némi lazasággal a pont szó helyett a kurzor szót is használhatjuk.

A számkapcsoló

Több parancsnak számkapcsolót adhatunk át, ami megmondja neki, hogy hányszor ismétlje meg ugyanazt. Például a *C-u 3 C-a* a harmadik sor elejére ugrik, a *C-u 3 C-e* ugyanennyit, de a sor végére. Üssük be még a *C-u 6 C-n* és a *C-u 6 C-p* parancsokat is, figyelve arra, hogy ugyanúgy viselkedik-e a kurzor, mint az előző példában, azaz a pont elé vagy a pont utánra ugrik-e! A *C-u 8 C-v* nem nyolc képernyőnyit, hanem csak nyolc sornyt mozgat visszafelé! A *C-u 8 M-v* szintén csak nyolc sort megy előre. A *C-u* előtag *u* betűje az *universal argument* szavakra utal, és ez az előtag adja át a számkapcsolókat az utána következő parancsoknak. Ha nem adunk meg számot a *C-u* után, az alapértelmezett érték mindig négy lesz, például a *C-u C-n* négygel fogja lefelé mozgatni a kurzort. Próbáljuk ki a *C-u C-l* parancsot különböző számkapcsolókkal, majd a *C-u* - billentyűkombinációt is!

Keresés és csere

Az *Edit > Search...* vagy a *C-s* indítja a keresést. Ahogy a keresendő szót a mini átmeneti tárolóba gépeljük, az Emacs rögtön ráugrik a már beírt karakterlánc első előfordulására. Hosszabb szavak esetén előfordulhat, hogy a teljes szót be sem kell írni ahhoz, hogy rögtön célhoz érjünk. Ha a megtalált szó újabb előfordulásaira is kíváncsiak vagyunk, akkor ismételten meg kell nyomnunk a *C-s* kombinációt. Az ENTER leütése jelzi a keresés befejeztét, de *C-s C-s* még ezután is

visszahívhatja a legutóbb keresett szót. A *C-g* kiléptet bennünket a keresésből, és a kurzort visszaviszi

a kiindulási helyre, az Esc szintén abba hagyatja a keresést, de a kurzort a megtalált szó helyén hagyja. Gépeléskor a DELETE megnyomása letörli a mini átmeneti tárolóba beírt karaktereket, és sorban visszaugrál a kiindulási helyre. Ha a keresett karakterlánc nem található a vizsgált átmeneti tárban, akkor figyelmeztető hangjelzést kapunk, és a mini átmeneti tárolóban megjelenik a *Failed I-search* üzenet. Az *Edit > Search backward...* vagy a *C-r* visszafelé keres, különben ugyanúgy viselkedik, mint a *C-s*. Szavak sorozatát a *C-s RET C-w szavak RET* vagy visszafelé a *C-r RET C-w szavak RET* parancsokkal tudjuk keresni. A keresett szavak közé egyetlen szóközt teszünk. Az ilyen kereséskor nem számít, hogy a szövegben lévő szavakat hány szóköz választja el egymástól, hogy van-e közöttük új sor vagy bármilyen már elválasztó karakter a szövegben. Lehetőség van reguláris kifejezések előre- és visszafelé történő keresésére is: a *C-M-s* és *C-M-r* billentyűkkel. A keresést és a helyettesítést az *M-x replace-string RET karakterlánc RET új karakterlánc RET* parancssal végezhetjük, ami *új karakterláncra* cseréli *karakterlánc* minden előfordulását. Reguláris kifejezések esetében az *M-x replace-regexp RET regexp RET új karakterlánc RET* formát használjuk. Ezek a helyettesítőparancsok kérdezés nélkül cserélik le a karakterláncokat. Ha azt akarjuk, hogy a program egyenként kérdezzen rá a csere szükségességére, akkor az *M-x query-replace RET karakterlánc<vége> RET új karakterlánc<vége> RET* alakot, reguláris kifejezések használatakor pedig az *M-x query-replace-regexp RET >regexp<vége> RET új karakterlánc<vége> RET* parancsot írjuk be. Ha az átmeneti tárban lévő összes előfordulást le akarjuk cserélni, előrefelé keresés esetén a szöveg elejére kell mennünk, mivel a keresés és a csere mindig onnan kezdődik, ahol a kurzor áll. Ha visszafelé haladunk, ilyenkor a szöveg végére kell ugranunk.

1. táblázat

	Görgetés balra		Görgetés jobbra
Előző képernyő		M-v;	
	C-x-<	C-l	C-x->
Következő képernyő		C-v	

2. táblázat

	függvény	mondat	sor	Kifejezés	szó	karakter		karakter	szó	kifejezés	sor	mondat	függvény
átmeneti tár								M-<					
oldal								C-x-[
bekezdés								M-{					
sor								C-p					
	C-M-a	M-a	C-a	C-M-b	M-b	C-b	pont	C-f	M-f	C-M-f	C-e	M-e	C-M-e
sor								C-n					
bekezdés								M-}					
oldal								C-x-]					
átmeneti tár								M->					

3. táblázat

	mondat	kifejezés	sor	szó	karakter	pont	karakter	szó	sor	kifejezés	mondat
Emacs	C-x Del	M-- C-M-k	M-O C-k	M-Del	Del?		C-d	M-d	C-k	C-M-k	M-k
XEmacs	C-x Del?	u.a.	u.a.	M-backspace	backspace		u.a.	u.a.	u.a.	u.a.	u.a.

Lapozgatás közben az állapotsoron láthatjuk, hogy a dokumentum hány százalékánál tartunk, de az *M-x line-number-mode RET* paranccsal ugyanott kiíratathatjuk, hogy a kurzor hányadik sorban jár. A parancs ismételt kiadásáig a sorszám mindig olvasható lesz. Az *M-x goto-line RET* a megadott sorra ugrik, a rövid alakja *M-g*.

Kijelölés

Ha az Emacs átmeneti tárában lévő szövegre a jobb egérgombbal rákattintunk, majd a gombot nyomva tartva elmozgatjuk az egeret, kijelölhetünk egy területet (region). A terület mindaddig kijelölve marad, amíg új kijelölést nem kezdeményezünk, vagy bele nem kattintunk a szövegbe. Kijelölhetünk egy területet úgy is, hogy a *C-szóköz* vagy a *C-@* kombinációval egy jelet (mark) teszünk a kurzor helyére, majd a kurzort elmozgatva kijelöljük a kiválasztandó területet. Nemcsak a kurzormozgató billentyűket használhatjuk, hanem a fent megismert kurzormozgató billentyűkombinációkat is, mint például *M-a*. Gondolom, nem lesz meglepő, hogy a *C-g* megnyomásával a kijelölést megszüntethetjük. Bizonyos műveletek csak a kijelölt területre hatnak. Tegyük egy jelet a kijelölés elejére a *C-szóközzel*, majd menjünk a kijelölés végére. Miután végeztünk a kívánt terület kijelölésével, a *C-x C-x* megnyomásával visszaugorhatunk a terület elejére, majd a *C-x C-u* parancs kiadásával az egész szöveget egy lépésben nagybetűssé alakíthatjuk. Most csináljunk egy újabb kijelölést, és a *C-x C-l* segítségével a szöveget alakítsuk vissza kisbetűssé! Hasznos lehet a *C-x C-x* akkor is, amikor elfelejtjük, hogy mi volt a kijelölés eleje vagy vége.

Ha egy kijelölésjelet a pillanatnyi kurzorállásra helyezzünk, majd megnyomjuk az *M-<* vagy *M->* billentyűket, az átmeneti tároló elejéig vagy végéig egy lépésben kijelölhetjük a szöveget. A *C-x h* az egész átmeneti tárat, a *C-x C-p* egy oldalt, az *M-h* egy bekezdést, *C-M-@* egy kifejezést, *C-M-h* pedig egy függvényt jelöl ki egy lépésben.

Ha nem az Emacs-on belüli átmeneti tárolók közt akarunk szöveget cserélni, a bal egérgombbal ki kell jelölnünk a

kívánt területet, ami rögtön X-kiválasztássá válik. Ez már minden további nélkül beszúrható lesz más X-szerkesztőbe: mint például a mostani cikk megírásához használt StarOffice 5.2-be. Ebben a szövegszerkesztőben a *Ctrl-v* billentyűkombinációt, az *Edit > Paste* menüt, vagy a beszúrást ikonként kell a beillesztéshez használni, de a KWrite vagy az Xcoral erre a célra a középső egérgomb megnyomását is elfogadja. Ha programozás közben C módban egy függvény törzsében kiadjuk az *M-C-h* parancsot, akkor egy lépésben kijelölhetjük az egész függvényt, mivel a parancs megkeresi a függvény elejét és a végét, majd aktívvá teszi.

Kivágás, másolás és beszúrás

A kivágást, a másolást és a beszúrást az ismert módon végezhetjük el: az Edit menüből vagy az idevágó ikonok használatával (1 g kép). De kifizetődő lesz elsajátítani az Emacs rövidítéseit is:

- C-w* – kivágja a kijelölt területet (wipe).
- M-w* – másolatot készít a kijelölt területről (wipe).
- C-y* – beilleszti a korábban kivágott vagy átmásolt területet (yank).

A *w* betű itt a wipe (töröl) szó helyett áll. Az Emacsban kétféle törlés van. Az egyik az *irtás* (kill), ami egy szövegrészt töröl, de egyúttal menti is az *Irtó gyűrűbe* (Kill Ring), ahonnan később előránthatjuk (yank), és beszúrhatjuk a kívánt helyre. Innen az *y* rövidítés. Egyetlen *Irtó gyűrű* létezik, és minden kiírtott szövegrész ide kerül, hogy később bármelyik puffer hozzáférhessen. A másik törlési mód megsemmisíti (*delete*) a szöveget, ami még egy ideig a visszaállító (*Undo*) paranccsal visszahozható, de utána véglegesen elvész. Ilyen megsemmisítő parancsot adunk ki például, amikor megnyomjuk a DELETE vagy a BACKSPACE billentyűket. Szintén mindent megsemmisítően töröl a kurzortól visszafelé a sor elejéig az *M-O C-k*, a *C-k* pedig a sor végéig. Ha a sor végén lévő új sor (newline) karaktert is törölni akarjuk, akkor a *C-u 1 C-k* parancsot kell kiadnunk, mert a *C-k* nem teszi meg. Ha beütjük a *C-u 1 C-k* parancsot,

láthatjuk, hogy az alul lévő sorok eggyel feljebb lépnek.

A 3. táblázat a megsemmisítő billentyűkombinációkat tartalmazza. Úgy vettem észre, hogy a törléseknél az Emacs és az XEmacs viselkedése nem mindenben feleltethető meg egymásnak. A leírásokkal ellentétben a *Del* például nálam nem töröl visszafelé, hanem ugyanúgy viselkedik, mint a *C-d*. Az Emacsban *C-x Del* valóban törli a mondatokat visszafelé, az XEmacsban viszont ugyanazt csinálja, mint az *M-k*. A táblázatban látható kérdőjelek erre utalnak. Programozás közben hasznos lehet, ha sorokat szűrhetünk be az adott sor mögé (*C-o*), vagy ha egyszerre távolíthatjuk el a szerkesztett sor mögötti, fölösleges, csak szóközüket tartalmazó, üres sorokat (*C-x C-o*). Ez utóbbi szintén megsemmisítő törlés.

Az *M-y* a legutóbbi beszúrást lecseréli az előző irtással. Ennek a parancsnak megint csak az *Irtó gyűrű* működésének ismeretében érthetjük meg az értelmét. Tudjuk, hogy az *Irtó gyűrű* eltárolja a legutóbbi harminc irtást, amit később az *M-y* segítségével sorban elővehetünk. Hogy értsük, miről van szó, kezdjünk el dolgozni az XEmacsban, közben a *C-w* vagy *M-w* parancsokkal töröljünk vagy másoljunk le kijelölt területeket! Ugorjunk az átmeneti tár elejére, és a *C-y* paranccsal szűrjük be a legutóbbi irtást! Ha most többször kiadjuk az *M-y* parancsot, az *Irtó gyűrűből* egyenként „előránthatjuk” az oda mentett területeket mindaddig, amíg meg nem találjuk a nekünk tetszőt. Láthatjuk tehát, hogy az *M-y* használata is egyfajta visszavonás (*undo*).

Sorozatunk következő részében a regiszterekkel, betűátalakítókkal és a könyvjelzőkkel ismerkedhetünk meg.



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.