

Emacs za začetnike

Jeremy D. Zawodny, jzawodn@wcnet.org, Prevod Andrej Grauf, andrej.grauf@amis.net v1.7, 14. oktober 1999, 22. april 2000

Ta dokument uvaja uporabnike sistema Linux v urejevalnik Emacs. Predpostavljam, da imate vsaj minimalno poznavanje urejevalnika vi oz. podobnih urejevalnikov. Zadnjo različico tega dokumenta lahko običajno najdete na naslovu <http://www.wcnet.org/jzawodn/emacs/>

Kazalo

1	Uvod	3
1.1	Avtorske pravice	3
1.2	Ciljna publika in namen	3
1.3	Kaj je Emacs?	3
1.3.1	Različice	4
1.3.2	Pridobivanje urejevalnika Emacs	4
2	Zagon Emacsa	4
2.1	Zagon & zaustavljanje Emacsa	4
2.1.1	Kaj boste videli?	4
2.2	Nekaj izrazov	5
2.2.1	Bufferji & datoteke	5
2.2.2	Točka (point) & področje (region)	6
2.2.3	Okna	6
2.2.4	Okvirji	6
2.3	Tipkovnica - osnove	6
2.3.1	Ukazne tipke (Meta, Esc, Control in Alt)	6
2.3.2	Premikanje po Bufferju	7
2.3.3	Bistveni ukazi	7
2.3.4	Dopolnjevanje ukazov s pomočjo tipke Tab	8
2.4	Vadnica, pomoč, & informacije	8
3	Načini urejevalnika Emacs	9
3.1	Načini Major v primerjavi z načini Minor	9
3.2	Programski načini	9
3.2.1	C/C++/Java	10
3.2.2	Perl	10
3.2.3	Python	10

3.2.4	Ostali	10
3.3	Avtorstvo	11
3.3.1	Preverjanje črkovanja (način ispell)	11
3.3.2	HTML (način html-helper)	11
3.3.3	TeX (tex-mode)	11
3.3.4	SGML (sgml-mode)	11
3.4	Ostali načini	11
3.4.1	Nadzor različic (vc mode)	12
3.4.2	Način lupine	12
3.4.3	Telnet in FTP	12
3.4.4	Man	12
3.4.5	Ange-FTP	12
4	Prilagajanje urejevalnika Emacs	13
4.1	Začasne prilagoditve	13
4.1.1	Dodeljevanje spremenljivk	13
4.1.2	Povezanost datotek	14
4.2	Uporaba datoteke .emacs	14
4.3	Prilagoditveni paket	15
4.4	Prikaz v X Windows	15
5	Priljubljeni paketi	15
5.1	VM (Pošta)	16
5.2	Gnus (pošta in novice)	16
5.3	BBDB (rollodex)	16
5.4	AucTeX (še en način TeX)	16
6	Ostali viri	17
6.1	Knjige	17
6.1.1	Učenje uporabe urejevalnika GNU Emacs	17
6.1.2	Pisanje razširitev za urejevalnik GNU Emacs	17
6.1.3	Programiranje v Lispu: Uvod	17
6.1.4	Referenčni priročnik za programski jezik Lisp v urejevalniku GNU Emacs	17
6.2	Spletne strani	18
6.2.1	EMACSLution	18
6.3	Novičarske skupine	18
6.4	Poštni sezname	18
6.5	Arhiv jezika Lisp za Emacs	18

1 Uvod

1.1 Avtorske pravice

Avtorske pravice © 1998 - 1999 Jeremy D. Zawodny. Razširjanje in spreminjanje tega dokumenta je dovoljeno pod pogoji navedenimi v Splošnem dovoljenju GNU. Izvod na zvezi lahko vedno najdete na <http://www.gnu.org/copyleft/gpl.html>

1.2 Ciljna publika in namen

Ta dokument je namenjen uporabnikom sistema Linux, ki bi se radi naučili nekaj malega o programu Emacs in ga tudi preizkusili. Prvotno je bil zasnovan kot osnutek kratkega tečaja, ki naj bi ga vodil na srečanju skupine uporabnikov sistema Linux iz okolice Toleda: <http://www.talug.org/>. Od takrat pa se je zaradi kopice koristnih povratnih informacij, ki sem jih prejel od skupine, nekoliko razširil. Več informacij lahko najdete v razdelku Zasluge.

V tem dokumentu ni nič, kar bi bilo namenjeno izključno sistemu Linux. Nanaša se na vse različice sistema Unix in celo za Emacs, ki deluje v sistemu Microsoft Windows. Ker pa je dokument del projekta dokumentacije za sistem Linux, naj poudarim, da je bil razvit za uporabnike sistema Linux.

Tisti, ki imate raje ime GNU/Linux kot samo "Linux" (preberite <http://www.gnu.org/gnu/linux-and-gnu.html> in videli boste, zakaj sta možni obe različici imen) lahko mirne volje vsa imena za sistem Linux, ki se pojavijo v tem dokumentu zamenjate z imenom GNU/Linux. Čeprav se strinjam z utemeljitvijo in dobro namerno idejo, se mi zdi popolnoma v redu uporabljati izraz GNU/Linux.

1.3 Kaj je Emacs?

Emacs različnim ljudem predstavlja različne stvari. Odvisno od tega, koga boste vprašali, lahko dobite naslednje odgovore:

- Urejevalnik teksta
- Odjemalec pošte
- Program za branje novic
- Word Processor
- Religija
- Integrirano razvojno okolje
- Karkoli pač želite da je!

Predpostavili bomo, da gre za urejevalnik teksta - neverjetno prilagodljiv urejevalnik teksta. V to vprašanje se bomo poglobili še nekoliko kasneje. Avtor urejevalnika Emacs je Richard Stallman (ustanovitelj Free Software Foundation: <http://www.fsf.org/> in projekta GNU <http://www.gnu.org/>), ki ga vzdržuje še danes.

Emacs je eden izmed najbolj priljubljenih in najbolj zmogljivih urejevalnikov teksta, ki se uporablja v sistemu Linux (in Unix). Po priljubljenosti zaostaja samo za urejevalnikom **vi**. Poznan je po svojem bogatem naboru funkcij, sposobnosti za enostavno prilagajanje in majhnemu številu napak. Velik nabor funkcij in sposobnost prilagajanja sta rezultat njegove zasnove in vgradnje. Ne da bi se spuščal v podrobnosti, bom povedal, da Emacs ni "samo urejevalnik". Je

urejevalnik, v večini napisan v programskem jeziku **Lisp**. Jedro urejevalnika Emacs predstavlja prevajalnik Lispa napisan v jeziku C. Samo najosnovnejši deli Emacs so napisani v jeziku C. Večina urejevalnika je dejansko napisana v programskem jeziku Lisp. Torej, v Emacs je na nek način “vgrajen” celoten programski jezik, ki ga lahko prilagajate, dopolnjujete in spreminjate njegovo obnašanje.

Emacs je tudi eden izmed najstarejših urejevalnikov. Dejstvo, da ga je v zadnjih 20-ih (?) letih uporabljalo na tisoče programerjev, kaže na to, da je na voljo veliko dodatnih paketov. Ti dodatki vam omogočajo početi z Emacsom stvari, o katerih Stallman verjetno sploh nikoli ni sanjal, ko je pričel delati na Emacsu. Več o tem v nadaljevanju.

Obstaja veliko spletnih strani in dokumentacije, ki vam nudi boljši pregled o urejevalniku Emacs, njegovi zgodovini in temah povezanih z njim. Teh stvari ne nameravam ponavljati in vam predlagam, da si ogledate razdelek 6 (Ostali viri) tega dokumenta.

1.3.1 Različice

Dobro je, če povem, da obstajata dve različici urejevalnika Emacs: GNU Emacs in XEmacs. Oba izhajata iz skupnega prednika in si delita večino funkcij. V tem dokumentu se bomo osredotočili na GNU Emacs (specifično, na različico 20.3) večina stvari, ki jih boste tu prebrali, pa se ravno tako nanaša na XEmacs in prejšnje različice GNU Emacs. V dokumentu bom to različico enostavno imenoval z “Emacs”.

1.3.2 Pridobivanje urejevalnika Emacs

Emacs lahko dobite zelo enostavno. Če uporabljate priljubljene distribucije Linuxa, kot so Debian, RedHat, Slackware, ali katero koli drugo, je Emacs verjetno dodan kot paket, ki ga lahko namestite iz vašega distribucijskega medija. Če ga na tem mediju ni, lahko dobite njegovo izvirno kodo in ga sami prevedete. Obiščite spletno stran GNU, kjer boste našli točno lokacijo: <http://www.gnu.org/software/emacs/emacs.html>

2 Zagon Emacs

2.1 Zagon & zaustavljanje Emacs

Kot nov uporabnik si verjetno želite zagnati Emacs, da bi se malce poigrali in preskusili stvari. Ko boste enkrat v urejevalniku in ga boste želeli zapustiti, vam mogoče to ne bo uspelo. Če Emacs niste še nikoli uporabljali, to storite sedaj. V pozorniku ukazne lupine vnesite `emacs` in pritisnite Enter. Emacs bi se moral zagnati. Če se ne, potem ni nameščen ali pa ni v vaši poti.

Ko ga boste usposobili, boste morali vedeti tudi, kako ga zapustiti. Kombinacija tipk za izhod iz Emacs je `C-x C-c`. Notacija `C-x` pomeni, da morate držati tipko `Ctrl` in potem pritisniti tipko `x`. V tem primeru boste morali pritisniti in držati tipko `Ctrl` in za zaključitev naloge pritisniti še tipko `c`.

Verjetno se vam bodo na začetku kombinacije tipk v Emacsu zdele čudne, tuje in mogoče celo neprijetne –še posebej če ste prej uporabljali urejevalnik `vi`. V nasprotju z urejevalnikom `vi`, Emacs nima različnih načinov za urejanje teksta in izdajanje ukazov.

Če ponovimo: z ukazom `emacs` boste zagnali Emacs. S kombinacijo tipk `C-x C-c` pa ga boste zaključili.

2.1.1 Kaj boste videli?

Ko boste zagnali Emacs, bo zasedel celotno okno X (oz. zaslon, če namesto sistema X Windows uporabljate konzolo). Na vrhu boste videli menije, v glavnem delu zaslona nekaj teksta, na dnu zaslona pa nekaj vrstic.

Izgledal bo nekako tako, kot tale skica tipa ASCII:

```
+-----+
|Buffers Files Tools Edit Search Mule Help|
|                                         |
|Welcome to GNU Emacs, one component of a Linux-based GNU system.|
|                                         |
|                                         |
|...                                     |
|---l:---Fl  *scratch*                  (Lisp Interaction)--Ll--All-----|
|For information about the GNU Project and its goals, type C-h C-p.  |
+-----+
```

OPOMBA: Emacs bo običajno zapolnil celoten zaslon/okno. V zgornjem primeru sem stvari malce stisnil, da bi prihranil na prostoru. Ko boste urejevalnik prvič zagnali, boste videli tudi pozdravno sporočilo. Tudi to sporočilo sem izpustil in ga na sliki zamenjal z "...". Pozdravno sporočilo vsebuje podatke o različici urejevalnika Emacs, ki ga uporabljate in vas opozarja na pomoč na zvezi in sorodno dokumentacijo.

Vrstica z meniji Najvišjo vrstico vmesnika v urejevalniku Emacs predstavlja meni. Če uporabljate sistem X, boste to vrstico prepoznali kot običajne spustne menije, ki jih lahko uporabljate s pomočjo miške. Drugače boste morali za dostop do njih uporabljati bližnjice na tipkovnici (ta dokument jih ne navaja).

Statusna vrstica in Mini-buffer Predzadnja vrstica v Emacsovem vmesniku je statusna vrstica. Vsebuje informacije o bufferju v katerem delate, v katerem načinu je Emacs in še mnoge druge stvari. Za zdaj bo dovolj že to, da veste, da obstaja.

Najnižja vrstica se imenuje **mini-buffer**. Od glavnega bufferja je ločena s pomočjo statusne vrstice, ki smo jo pravkar spoznali. Mini-buffer si lahko predstavljate kot Emacsovo "ukazno vrstico". Na tem mestu se izpisujejo vsi ukazi, ki jih daste urejevalniku, pa tudi statusna sporočila, kot odgovori na vaše početje.

V dokumentaciji za Emacs se stvar, ki jo jaz imenujem statusna vrstica običajno imenuje načinovna vrstica (mode line). V njej Emacs prikazuje sporočila o trenutnem načinu, ki ga uporabljate, pa tudi druge stvari, kot je na primer trenutni datum in čas, številka vrstice, velikost datoteke in skoraj vsi drugi podatki, ki jih želite videti.

2.2 Nekaj izrazov

V tem razdelku boste našli najosnovnejše izraze na katere boste naleteli pri delu z urejevalnikom Emacs in branju dokumentacije.

2.2.1 Bufferji & datoteke

V nasprotju z nekaterimi urejevalniki v Emacsu, ko odprete datoteko, le ta ne ostane "odprta" ves čas, ko z njo delate. Namesto tega jo Emacs prebere v **buffer** v pomnilniku. Medtem ko urejate buffer in delate s podatki, se na disku nič ne spremeni. Datoteka se na disku posodobi samo takrat, ko buffer dejansko shranite. Seveda obstajajo prednosti in pomankljivosti takšnega načina, vendar je za vas pomembno le, da veste da stvar deluje na ta način.

Kot posledico takšnega shranjevanja boste opazili, da se v dokumentaciji za urejevalnik Emacs, v načinih in paketih uporablja izraz "buffer". Zapomnite si, da buffer pomeni "kopijo datoteke, ki je trenutno v pomnilniku." Ah ja, da ne pozabim, ni nujno, da se buffer vedno nanaša na določeno datoteko na disku. Pogosto bo Emacs tvoril bufferje kot rezultat ukazov, ki ste jih zagnali. Takšni bufferji lahko vsebujejo rezultate ukaza, sezname s katerih lahko izbirate itd.

2.2.2 Točka (point) & področje (region)

V jeziku Emacsa boste pogosto slišali oz. videli reference za **točko**. V splošnem točka predstavlja kazalec. Ko boste pričeli uporabljati urejevalnik Emacs, razlika med točko in kazalcem verjetno še ne bo pomembna. Če pa ste radovedni, si lahko razliko predstavljate na nasledni način. Kazalec je vizualna predstavitev točke. Kazalec je vedno “na” določenem znakovnem položaju v trenutnem bufferju, točka pa je v prostoru *med dvema znakoma* v bufferju. Torej lahko rečemo, če je kazalec v besedi “Linux” na črki ‘n’, je potem točka med črko ‘i’ in ‘n’.

Kot večina sodobnih urejevalnikov, tudi Emacs omogoča izvajanje operacij (poravnava, črkovanje, preoblikovanje, izrezovanje, kopiranje, lepljenje, ...) na razdelku v trenutnem bufferju. Blok besedila lahko osvetljite (oz. “označite”) s pomočjo tipkovnice ali miške in potem izvršite določeno operacijo samo na izbranem besedilu. V urejevalniku Emacs se blok besedila imenuje **področje**.

2.2.3 Okna

Ta del bo nekoliko begajoč za vse tiste, ki so kdaj uporabljali vmesnik GUI. Zapomnite si samo, da je bil Emacs razvit veliko pred vmesniki GUI in še preden so postali priljubljeni upravljalniki oken.

Okno v urejevalniku Emacs predstavlja področje zaslona v katerem je prikazan buffer. Ko boste Emacs prvič zagnali, boste videli na zaslonu samo eno okno. Nekatere funkcije Emacsa (kot je na primer pomoč in dokumentacija) pogosto [začasno] odprejo dodatna okna na zaslonu Emacsa.

Okna v urejevalniku Emacs nimajo nič skupnega z X Windows v pomenu GUI. V Emacsu lahko nove bufferje odprete tako, da odprete dodatna okna X. Na ta način lahko primerjate dve datoteki eno ob drugi. Ta okna X se v jeziku Emacs imenujejo **okvirji**. Berite dalje.

2.2.4 Okvirji

V urejevalniku Emacs, je **okvir** ločeno, samostojno okno X v katerem je prikazan buffer urejevalnika Emacs. Oba pa sta del iste seje urejevalnika Emacs. Obnašanje je nekoliko (vendar ne preveč) podobno, kot če bi v Netscape Navigatorju pritisnili kombinacijo tipk Alt+N.

2.3 Tipkovnica - osnove

Ta razdelek vsebuje osnove, ki se nanašajo na delovanje tipkovnice za Emacs. Kot v vsakem zmogljivem urejevalniku lahko tudi v urejevalniku Emacs vse stvari izvršite s samo nekaj pritiski na tipke.

Če ste uporabnik urejevalnika vi, potem ste verjetno potrebovali nekoliko časa, da ste se navadili na tipke k, j, l, h za premikanje ene vrstice navzgor, navzdol, za en znak naprej in en znak nazaj. Mogoče ste porabili nekaj ur ali celo tednov, da ste se privadili navigacije po datoteki zaradi velikega števila kombinacij tipk, ki so na voljo v urejevalniku vi.

Tudi Emacs ni nič kaj drugačen. Naučiti se bo treba različnih kombinacij tipk in ukazov. Prav tako, kot v urejevalniku vi, se boste morali za opravljanje večine dela naučiti samo osnov. Sčasoma pa boste lahko razširili svoje znanje in poiskali hitrejša pota za opravljanje stvari.

2.3.1 Ukazne tipke (Meta, Esc, Control in Alt)

Kot boste kmalu spoznali, se v Emacsu zelo veliko uporabljajo kombinacije različnih tipk. Ker ni načinovni urejevalnik tako kot vi, vam ni potrebno paziti, ali ste v “ukaznem načinu” ali “načinu za urejanje” preden boste poskusili premakniti kazalec in izvršiti ukaz. Namesto tega pritisnete samo pravo kombinacijo tipk, Emacs pa bo (običajno) izvršil nalogo, ki ste mu jo zadali.

Za tipke, ki se najpogosteje uporabljajo v urejevalniku Emacs, se v dokumentaciji uporabljajo okrajšave, na primer C (za Control oz. Ctrl) in M za (Meta). Čeprav ima večina sodobnih tipkovnic za osebni računalnik eno ali več tipk označeno s Ctrl, jih ima le malo tipko z oznako Meta. Verjetno boste hoteli zamenjati tipko Esc ali Alt za tipko Meta. V večini standardnih nastavitvah pa tako tipka Esc kot tudi tipka Alt služita enakemu namenu.

Ko boste torej v dokumentaciji za urejevalnik Emacs videli zapis C-x f to pomeni "pritisni control-x in potem še tipko f." Če pa boste videli zapis M-x shell, to pomeni "pritisni alt-x in potem vtipkaj besedo shell".

Zelo koristen ukaz za začetnike je M-x apropos oz. C-h a. apropos bo poiskal v Emacsovi dokumentaciji na zvezi vse funkcije in regularne izraze, ki jih vi vnašate. To je odličen način za raziskovanje in spoznavanje vseh ukazov, ki se nanašajo na okvirje. Enostavno pritisnite kombinacijo tipk C-h a nato pa še frame.

2.3.2 Premikanje po Bufferju

Sedaj ko veste kaj pomenijo vse te prestižne okrajšave, si oglejmo še seznam najobičajnejših tipk, ki se uporabljajo za premikanje po bufferju:

Kombinacija tipk	Akcija

C-p	Eno vrstico navzgor
C-n	Eno vrstico navzdol
C-f	Naprej za en znak
C-b	Nazaj za en znak
C-a	Na začetek vrstice
C-e	Na konec vrstice
C-v	Eno stran navzdol
M-v	Eno stran navzgor
M-f	Naprej za eno besedo
M-b	Nazaj za eno besedo
M-<	Na začetek bufferja
M->	Na konec bufferja
C-g	Končaj trenutno operacijo

Tudi smerne tipke (oz. tipke s puščicami) običajno delujejo tako, kot od njih pričakujete. Mogoče ne bo delovala tipka Backspace. Vendar je to že druga zgodba. :-)

2.3.3 Bistveni ukazi

No dobro, sedaj veste kako se premikati znotraj bufferja, kaj pa odpiranje in shranjevanje datotek? Iskanje? Tukaj je nekaj osnovnih ukazov.

Preden preidemo na te ukaze, naj najprej na kratko pojasnim kako stvari sploh delujejo.

Vse "kombinacije ukazov" v urejevalniku Emacs (tiste, ki imajo obliko M-x nekaj ali C-nekaj) so dejansko samo bližnjice do funkcij, ki so del urejevalnika Emacs. Poljubno funkcijo lahko prikličete z vnosom kombinacije tipk M-x ime-funkcije in s pritiskom na tipko Enter. Uporabite lahko tudi bližnjico na tipkovnici za to funkcijo (če le obstaja).

Na primer, funkcija urejevalnika Emacs, ki shrani buffer na disk se imenuje save-buffer. Glede na privzeto vrednost je pripeta tudi na C-x C-s. Torej, za shranitev trenutnega bufferja lahko uporabite bližnjico ali pa vtipkate M-x save-buffer in v obeh primerih dosežete isti rezultat.

Vse najobičajnejše funkcije imajo glede na privzeto vrednost bližnjice na tipkovnici. Nekatere izmed njih so navedene v naslednji tabeli.

kombinacija tipk	Funkcija	Opis
C-x C-s	save-buffer	Shrani trenutni buffer na disk
C-x u	undo	Razveljavi zadnjo operacijo
C-c C-f	find-file	Odpri datoteko na disku
C-s	isearch-forward	Nadaljuj z iskanjem niza
C-r	isearch-backward	Išči niz od spodaj navzgor
	replace-string	Poišči & zamenjaj niz
	replace-regexp	Poišči & zamenjaj s pomočjo regexp
C-h t	help-with-tutorial	Uporabi interaktivno vadnico
C-h f	describe-function	Prikaži pomoč za funkcijo
C-h v	describe-variable	Prikaži pomoč za spremenljivko
C-h x	describe-key	Prikaži čemu služi zaporedje tipk
C-h a	apropos	Poišči pomoč za niz/regexp
C-h F	view-emacs-FAQ	Prikaži FAQ za Emacs
C-h i	info	Preberi dokumentacijo za Emacs
C-x r m	bookmark-set	Postavi oznako. Uporabno pri iskanu
C-x r b	bookmark-jump	Skoči na oznako.

Ko boste preizkusili večino teh funkcij, boste opazili da jih kar nekaj od vas zahteva vnos podatkov. Te podatke boste vedno vnesli v mini-buffer. To je podobno uporabi ukaza `:` v urejevalniku `vi` oz. večini ukazov, ki jih uporabljate v vaši lupini sistema Unix.

Emacs ima dobresedno na stotine vgrajenih funkcij. Zgoraj prikazan seznam je samo majhen del tistih funkcij, ki jih redno uporabljam. Popolnejši seznam funkcij, ki so na voljo boste našli v pomoči na zvezi. Tam boste našli tudi popolnejšo dokumentacijo za tiste funkcije, ki sem jih navedel v zgornji tabeli.

2.3.4 Dopolnjevanje ukazov s pomočjo tipke Tab

Kot večina priljubljenih lupin sistema Unix (`bash`, `csh`, `tcsh`, ...), tudi Emacs nudi dopolnjevanje ukazov s pomočjo tipke `Tab`. Pravzaprav je bilo dopolnjevanje ukazov v lupini `bash` oblikovano po tistem v urejevalniku Emacs. Če torej uporabljate to funkcijo v lupini `bash` vam ne bi smela delati preglavic v urejevalniku Emacs.

Kot primer, poskusite `M-x search` in nato pritisnite tipko `Tab`. Emacs bo pripel vezaj in s tem nakazal, da za dopolnjevanje obstaja več možnosti, vse možnosti pa kot naslednji znak vsebujejo vezaj. Pritisnite na tipko `Tab` še enkrat in Emacs bo prikazal seznam vseh možnih funkcij iz katerega lahko izberete ustrezno. Opazili boste, da to stori v *novem oknu*. Začasno razpolovi vaš zaslon na dva dela: en vsebuje buffer, ki ste ga urejali, drug pa vsebuje seznam vseh možnih dopolnitev za "search-". Proces za izbiro lahko prekinete s kombinacijo tipk `C-g`. S tem pa boste zaprli tudi novo okno.

2.4 Vadnica, pomoč, & informacije

Emacs vsebuje vadnico na zvezi, ki vas vodi skozi osnovne funkcije za urejanje in funkcije, ki bi jih moral vsakdo poznati. Pravtako boste našli razlago, kako uporabiti ostale funkcije pomoči v urejevalniku Emacs.

Toplo vam priporočam, da porabite nekaj časa in greste skozi vadnico, seveda če se mislite resno naučiti uporabe urejevalnika Emacs. Kot je prikazano v zgornji tabeli, lahko pričnete z vadnico preko kombinacije tipk `C-h t`. Vadnica je namenjena tistim ljudem, ki so šele pričeli z delom v urejevalniku Emacs.

Če uporabljate Emacs v sistemu X, boste opazili da se skrajno desni meni v vrstici z meniji imenuje Pomoč (Help). Ko boste raziskovali ta meni, boste verjetno opazili, da imajo nekatere funkcije na desni strani pripisane še bližnjice.

Da bi si ogledali celotno dokumentacijo, ki je na voljo v urejevalniku Emacs, preskusite kombinacijo tipk `M-x info` ali `C-h i`, s čimer boste zagnali Emacsov brskalnik po dokumentaciji.

3 Načini urejevalnika Emacs

Načini v urejevalniku Emacs predstavljajo različna obnašanja in lastnosti, ki jih lahko vključite in izključite (ali prilagodite) v različnih okoliščinah. Prav ti različni načini naredijo urejevalnik (Emacs) primeren za pisanje dokumentacije, za programiranje v različnih jezikih (C, C++, Perl, Python, Java, in še v veliko drugih), za tvorjenje domačih strani, pošiljanje elektronskih sporočil, branje novic Usenet, vodenje vaših sestankov in celo za igranje iger.

Načini v Emacsu so dejansko samo knjižnice kode v programskem jeziku Lisp, ki na nek način razširijo, prilagodijo ali posodobijo urejevalnik Emacs.

3.1 Načini Major v primerjavi z načini Minor

Primarno sta na voljo dva različna tipa načinov: Major in Minor. Dokler ne boste pričeli delati z njimi, boste razliko dokaj težko razumeli. Pa vseeno poskusimo.

V določenem trenutku je lahko aktiven samo en način major, medtem ko je lahko v določenem trenutku aktivnih več načinov minor. Načini tipa major so običajno specifični za določeno nalogo oz. programski jezik, medtem ko so načini minor manjši in manj specifični pripomočki, ki izpolnjujejo veliko opravil.

To zveni nekoliko abstraktno, pa poskusimo razložiti s primerom. Obstaja način, ki ga zelo pogosto uporabljam, kadar pišem preproste stare tekstovne datoteke. Imenuje se tekstovni način (`text-mode`). Ta način je bil razvit za pisanje prosto oblikovanega besedila, na primer za datoteke README. Ta način zna prepoznavati besede in odstavke in običajno preverja, da počne stvari, ki jih od njega pričakujem, kadar uporabljam običajne navigacijske tipke.

Kadar pa pišem besedila za ljudi, običajno želim, da dobro izgledajo. Besedila morajo biti lepo prelomljena. Da bi omogočil deljenje besed, samo vključim način tipa minor `auto-fill`. Ta način poskuša storiti Pravilne Stvari, kadar vnašam besedilo in pridem do konca vrstice. Dejstvo da gre za način tipa minor pomeni, da lahko deluje v različnih načinih tipa major. Definicija "Prave Stvari", ko dosežem konec vrstice se razlikuje kadar sem v tekstovnem načinu (`text-mode`) od definicije kadar sem na primer v java-načinu (`java-mode`). Ne želim da bi bila moja koda jezika Java deljena, kot da bi šlo za slovensko besedilo. Vsekakor pa *želim*, da so bloki komentarja v kodi jezika Java pravilno deljenji! Način `auto-fill` je dovolj prebrisan, da za te stvari samodejno poskrbi.

Avtorji številnih načinov v urejevalniku Emacs so se zelo potrudili, da stvari, ki bi morale delovati kot načini tipa minor tako tudi delujejo.

Če se vrnete nazaj in si ogledate skico ASCII zalona urejevalnika Emacs, boste opazili, da načinovna vrstica prikazuje načine v katerih je Emacs. V tistem primeru je bil to način z imenom "Lisp Interaction", ki je privzet način. Uporaben je samo v primeru, če boste želeli programirati v Lispu. (Zakaj pa ne, saj je večina urejevalnika Emacs napisana v tem jeziku?)

3.2 Programski načini

Prvo in najpomembnejše, Emacs je zgradil programer za programerje. Obstajajo zelo kvalitetni načini za skoraj vse priljubljene programske jezike (in celo za nekatere manj priljubljene). Na kratko bom opisal samo nekatere.

Večina programskih načinov ima nekaj skupnih karakteristik. Običajno izvajajo nekatere ali vse izmed naslednjih operacij:

- Nudijo barvno označevanje skladnje jezika.
- Nudijo samodejno poravnavo in oblikovanje kode programskega jezika.
- Nudijo pomoč, občutljivo na vsebino (jezik).
- Samodejno komuniciranje z vašim razhroščevalnikom.
- V vrstico z meniji dodajo menije, ki so specifični za posamezen jezik.

Obstaja tudi nekaj načinov, ki niso specifični za jezik, pomagajo pa pri izvajanju opravil, ki so običajni pri programiranju v večini jezikov. Stvari kot je komuniciranje z vašo programsko opremo za nadziranje različic, samodejno dodajanje komentarjev v vašo kodo, tvorjenje datotek Makefiles, posodabljanje dnevnikov, itd.

Ko seštejete vse te načine in dodate še zrelost in zanesljivost kode urejevalnika Emacs, ga je kaj mogoče primerjati s komercialnimi Integriranimi razvojnimi okolji (Integrated Development Environments - IDE) za jezike kot sta C++ in Java. Seveda pa je urejevalnik Emacs prosto dostopen.

3.2.1 C/C++/Java

Ker je skladnja jezikov C, C++, in Java dokaj podobna, obstaja en sam način urejevalnika Emacs, ki skrbi za vse tri jezike (pa tudi za Objective-C in IDL). Je zelo zrel in popoln paket in je vključen v distribucijo Emacs. Ta način pokličete s kombinacijo tipk `cc-mode` ali `CC Mode`.

Več podrobnosti ali če želite dobiti novo različico najdete na spletnem naslovu <http://www.python.org/emacs/>.

3.2.2 Perl

Dejansko obstajata dva načina za urejanje kode Perl v urejevalniku Emacs. Prvi način pokličete s kombinacijo tipk `perl-mode` (kot ste verjetno že sami ugotovili) drugega pa s kombinacijo `cperl-mode`. Ne vem zakaj obstajata dva načina (tega ne piše v dokumentaciji), zdi pa se, da je način `perl-mode` originalni način za urejanje kode Perl v urejevalniku Emacs. Vsebuje nekoliko manj lastnosti kot `cperl-mode`, manjka pa mu tudi sposobnost prepoznavanja nekaterih novejših konstruktov jezika Perl.

Jaz osebno uporabljam in vam priporočam `cperl-mode`, ki je dokaj aktivno vzdrževan in vsebuje vse lastnosti, ki sem jih kadarkoli želel. Zadnjo različico lahko najdete na naslovu: <ftp://ftp.math.ohio-state.edu/pub/users/ilya/emacs>.

Ne držite me za besedo. Preizkusite oba in izberite tistega, ki bolj ustreza vašim zahtevam.

3.2.3 Python

Tudi za Python (še en zelo priljubljen skriptni jezik) je v urejevalniku Emacs na voljo način. Kolikor vem, tega načina *ni* v distribuciji urejevalnika GNU Emacs, je pa priložen distribuciji urejevalnika XEmacs. Vendar dokaj dobro deluje v obeh urejevalnikih.

Način `python-mode` lahko najdete na uradni spletni strani Pythona <http://www.python.org/emacs/python-mode/>.

3.2.4 Ostali

Obstaja veliko veliko ostalih načinov za urejanje, ki pomagajo programerjem. Ti načini vam lahko pomagajo pri stvareh kot so:

- Skripti ukazne lupine (Bash, sh, ksh, csh, ...)
- Awk, Sed, Tcl, ...

- Datoteke Makefiles
- Dnevnik
- Dokumentacija
- razhroščevanje

In še veliko več. Oglejte si zadnji del te dokumentacije, kjer boste našli več informacij o dodatkih in iskanju drugih načinov.

3.3 Avtorstvo

Posebni načini urejevalnika Emacs *niso* omejeni samo na tiste ljudi, ki so jih napisali. Tudi ljudje, ki pišejo dokumentacijo (kakršno koli), lahko imajo koristi od velikega izbora načinov v urejevalniku Emacs.

3.3.1 Preverjanje črkovanja (način `ispell`)

Avtorji, ki pišejo različno dokumentacijo, morajo vsake toliko časa preveriti pravilnost črkovanja. Če imate nameščen **GNU ispell**, lahko uporabite kombinacijo tipk `M-x ispell` in preverite pravilnost črkovanja v trenutnem bufferu. Če ispell najde besede, ki jih ne pozna, vas bo o tem obvestil s pomočjo seznama možnih zamenjav in vam omogočil izbrati eno (ali nobeno) izmed njih. Njegova funkcionalnost se lahko kosa s črkovalniki v priljubljenih komercialnih paketih programske opreme.

3.3.2 HTML (način `html-helper`)

Če vsake toliko časa napišete kakšno datoteko tipa HTML (ali pa to počnete pogosto), boste mogoče preizkusili način `html-helper-mode`. Dobite ga lahko na naslovu <http://www.santafe.edu/~nelson/tools/>, kot tudi dokumentacijo in stvari, ki so povezane s tem načinom.

Kot je razvidno iz njegovega imena, način `html-helper-mode` nudi veliko stvari, ki pomagajo tistim ljudem, ki še vedno ročno pišejo dokumente tipa HTML -na staromodni način.

3.3.3 TeX (`tex-mode`)

Kadar pišete dokumente v urejevalniku TeX, je pogostokrat v pomoč, če si priskrbite urejevalnik Emacs, ki vnese nekoliko barv in nudi osvetljevanje oklepajev in ostalih znakov. Način `tex-mode` sam poskrbi za te stvari.

Čeprav ne pišem več pogostokrat neposredno v urejevalniku TeX, če pa to že počnem, ta način poskrbi za preglednejšo izvirno kodo TeX.

3.3.4 SGML (`sgml-mode`)

Dokument, ki ga sedaj berete je bil napisan v SGML (oz. pretvorjen v format, v katerem ga trenutno berete). `sgml-mode` nudi vse osnove za pisanje dokumentov tipa SGML: ovrednotenje (validation), osvetljevanje, forward-tag, backward-tag in še veliko več. Ta način je standarden del urejevalnika Emacs.

3.4 Ostali načini

Seveda obstaja še veliko drugih priročnih načinov, ki vam olajšajo delo. Naštel bom samo nekaj najbolj priljubljenih:

3.4.1 Nadzor različic (vc mode)

Način vc komunicira z veliko večino popularnih programov za nadzor različic (RCS, SCCS, CVS), kar zelo poenostavi označevanje in odznačevanje datotek, upravljanje z izdajami itd. Je standarden del urejevalnika Emacs, dokumentacijo pa lahko najdete v dokumentaciji urejevalnika Emacs.

3.4.2 Način lupine

Zakaj bi odpirali novo okno X ali navidezno konzolo samo zaradi tega, da bi izvršili nekaj ukazov v lupini? Storite to kar v urejevalniku Emacs in si prihranite nekaj težav. :-)

M-x shell bo znotraj bufferja urejevalnika Emacs zagnal lupino. V tem bufferju lahko počnete skoraj vse stvari, ki jih lahko počnete tudi v običajnem pozorniku lupine (razen zaganjanje programov kot je vi ali pine), ker se urejevalnik Emacs v ozadju pogovarja z vašo dejansko lupino.

Tudi to je standarden del urejevalnika Emacs in je dokumentiran v dokumentih urejevalnika Emacs.

3.4.3 Telnet in FTP

Zakaj bi odpirali novo okno X ali navidezno konzolo, samo da bi zagnali telnet ali FTP? Storite to kar v urejevalniku Emacs in si prihranite nekaj težav. (Ali ste že opazili, da se ponavljam?)

Prav tako kot zaganjanje lupine znotraj urejevalnika Emacs, lahko zaženete tudi telnet in ftp. Preizkusite M-x telnet ali M-x ftp in se prepričajte sami. Preberite dokumentacijo in poiščite vse podrobnosti.

3.4.4 Man

Zakaj bi prehajali v drugo okno X ali navidezno konzolo, samo da bi prebrali strani programa man? Storite to kar v urejevalniku Emacs in si prihranite nekaj težav. (Obljubim, da bom sedaj nehal s tem.)

Prav tako kot zaganjanje lupine znotraj urejevalnika Emacs, lahko prebirate tudi strani priročnika. Preiskusite M-x man in se prepričajte sami. Več podatkov lahko najdete v dokumentaciji.

3.4.5 Ange-FTP

Citat iz dokumentacije ange-ftp:

Ta paket poskuša olajšati dostopanje do datotek in imenikov, ki uporabljajo FTP znotraj urejevalnika GNU Emacs. Podmnožica teh običajnih rutin za delo z datotekami je razširjena na delo s programom FTP.

To pomeni, da lahko obravnavate datoteke, ki so na oddaljenih računalnikih, kot bi bile lokalne. Če torej morate urediti datoteko na drugem računalniku, samo povejte urejevalniku Emacs naj jo odpre (s pomočjo malce različne sintakse poti) in poskrbi za vse podrobnosti prijavljanja in prejemanja datoteke. Ko potem shranite datoteko s pomočjo C-x C-s, ange-ftp prestreže proces shranjevanja in datoteko zapiše nazaj na oddaljen računalnik.

Malce drugačna sintaksa poti se glasi... Datoteka z imenom "mojadatoteka", v imeniku "uporabnik", na računalniku z imenom "moj.gostitelj.org" se lahko odpre z odprtjem datoteke (C-x f):

/uporabnik@moj.gostitelj.org:~uporabnik/mojadatoteka

Tudi to je standardni del distribucije urejevalnika Emacs, torej lahko najdete informacije v dokumentaciji za urejevalnik Emacs.

Za zgornji primer se zahvaljujem Etiennu Grossmannu (*etienne@anonimo.isr.ist.utl.pt*).

4 Prilagajanje urejevalnika Emacs

Skoraj vse prilagoditve urejevalnika Emacs se vršijo s pomočjo spreminjanja kode Lispa. Spremenite lahko spremenljivke, ki vplivajo na način delovanja urejevalnika Emacs, lahko pa mu dodate tudi nove funkcije (ali prepišete obstoječe funkcije -jih zamenjate s svojimi).

4.1 Začasne prilagoditve

Ko boste eksperimentirali s prilagoditvami urejevalnika Emacs, boste verjetno želeli, da so te prilagoditve samo začasne. Če se kaj hudo zalomi, uporabite samo kombinacijo tipk `C-x C-c` za izhod iz urejevalnika Emacs in ga ponovno zaženete. Ko boste ugotovili, katere spremembe naj bodo trajne, jih lahko dodate v svojo različico datoteke `.emacs`, kjer se bodo ob zagonu urejevalnika Emacs vedno upoštevale. To bomo obravnavali v naslednjem razdelku.

4.1.1 Dodeljevanje spremenljivk

Prilagoditve boste najenostavneje izvršili tako, da boste spremenili vrednost spremenljivki v urejevalniku. Koda za takšno spremembo izgleda nekako tako:

```
(setq ime-spremenljivke nova-vrednost)
```

Kjer je `ime-spremenljivke` ime spremenljivke in `nova-vrednost` vrednost, ki bi jo radi dodelili spremenljivki. (Povedano v jeziku Lisp, spremenljivko boste pripeli na določeno vrednost.) Funkcija `setq` v jeziku Lisp je analogna dodeljevalnim operatorjem (običajno je to `=`) v drugih programskih jezikih.

OPOMBA: Izpustil sem veliko podrobnosti, da bi razlaga ostala karseda preprosta. Mogoče boste videli, da drugi ljudje uporabljajo tudi funkcijo programskega jezika Lisp `set` ali celo funkcijo `setq-default`. Če vas te funkcije res zanimajo, jih poiščite v referenčnem materialu za Lisp v urejevalniku Emacs.

Oglejmo si vrstico v moji datoteki `.emacs`

```
(setq-default transient-mark-mode t)
```

Spremenljivka `transient-mark-mode` nadzira ali postane področje osvetljeno, kadar ga označim. V mnogih aplikacijah GUI, kadar z miško izberete območje besedila, le to postane osvetljeno z neko barvo. Tudi Emacs bo storil tako, če bo nastavljena spremenljivka `transient-mark-mode` (na neničelno vrednost).

Na *KAKŠNO* vrednost?

Dobro. Če se le malce oddaljimo od teme. Večina programskih jezikov uporablja neko notacijo za vrednosti prav/narobe. V jeziku C/C++ se vrednost smatra za pravilno, če je neničelna vrednost. V jeziku Perl je pravilna neničelna vrednost. Tudi v jeziku Lisp velja isti dogovor, razlikujejo se samo imena in simboli.

Pravilno se običajno zapiše s `t`, nepravilno (oz nič) pa se zapiše z `nil`. Kot v drugih jezikih, se vsaka neničelna vrednost smatra za pravilno.

Da bi dobili popoln občutek o tem, čemu je način `transient-mark-mode` sploh namenjen, lahko uporabite pomoč na zvezi. Vnesite `C-h` v ali `M-x describe-variable` in potem `transient-mark-mode`. Če ste tako leni kot jaz, boste verjetno uporabili lastnost dopolnjevanja s pomočjo tipke `Tab`. Vtipkajte samo del imena spremenljivke in pritisnite na tipko `Tab`. Če boste vnesli dovolj imena in ga bo Emacs prepoznal kot unikatno ime, ga bo tudi dopolnil.

Naslednja spremenljivka, ki jo ljudje pogosto spreminjajo je `fill-column`. Le ta pove urejevalniku Emacs, kako širok naj bo zaslon za deljenje besed (in `auto-fill-mode` upošteva to vrednost). Vtipkate lahko neko nesmiselno vrednost:

```
(setq fill-column 20)
```

To pa ne bi delovalo. Urejevalniku Emacs morate povedati naj **ovrednosti** izraz, ki ste ga vnesli. To storite tako, da postavite točko (kazalec) na konec izraza in potem vnesete `C-x C-e`, s čimer boste poklicali funkcijo `eval-last-sexp` če vas seveda to gane. Ko boste to storili, boste lahko opazili, da se vrednost 20 (oz. katero vrednost ste pač vnesli) izpiše v mini-bufferju na dnu zaslona. To je samo vrnjena vrednost iz izraza, ki ste ga ovrednotili.

Da bi preskusili če zadeva deluje, vtipkajte stavek ali dva. Če imate vključen način `auto-fill-mode` (verjetno ga nimate), boste opazili da se besede delijo na dvajseti vrstici. Ko boste vnesli nekaj besedila, vtipkajte še `M-q`, s čimer boste poklicali funkcijo `fill-paragraph`. Le ta bo izvršila deljenje besed.

4.1.2 Povezanost datotek

Urejevalnik Emacs lahko prilagodite tako, da samodejno izvrši določene operacije, ko odprete določen tip datoteke (ravno tako, kot določeni GUI samodejno zaženejo specifično aplikacijo, ko kliknete na ikono določene datoteke). Želim si, da bi urejevalnik Emacs samodejno preklopi v način `text-mode` vsakič, ko odprem datoteko s končnico `.txt`. No, dejansko se to že dogaja. :-) Torej, povejmo urejevalniku Emacs naj vedno preide v način `text-mode`, kadar odprem datoteko z imenom "README".

```
(setq auto-mode-alist (cons '("README" . text-mode) auto-mode-alist))
```

Huh?

Ne da bi se spustili v podrobnosti programiranja v jeziku Lisp, ki vam jih ni potrebno videti (ne bo pa nič narobe, če se jih boste naučili), povejmo samo, da spremenljivka `auto-mode-alist` vsebuje seznam parov. Vsak par vsebuje regularni izraz in ime načina v urejevalniku. Če datoteka, ki jo odprete ustreza regularnemu izrazu (v tem primeru nizu `README`) bo urejevalnik Emacs zagnal način, ki ste ga podali.

Zgornjo skladnjo smo uporabili, ker smo na seznam dejansko dodali nov par. Verjetno si ne želite preiti na `auto-mode-alist`, ne da bi se prej prepričali, da se vrednosti, ki jih že vsebuje, niso izgubile.

Če pa želite, da Emacs samodejno preklopi na način `html-helper-mode` vsakič, ko odprete datoteke, ki se končajo s končnico `.html` ali `.htm`, bi v vašo datoteko `.emacs` dodali:

```
(setq auto-mode-alist (cons '("\\.html$" . html-helper-mode) auto-mode-alist))
(setq auto-mode-alist (cons '("\\.htm$" . html-helper-mode) auto-mode-alist))
```

Možnosti je res neskončno.

4.2 Uporaba datoteke `.emacs`

Ko boste že nekaj časa delali v urejevalniku Emacs in dobili nekaj občutka, kako vam lahko koristijo prilagoditve, boste verjetno želeli nekatere stvari narediti trajne (oz. vsaj dokler si ne boste premislili). Če boste Emacs vsakodnevno uporabljali, boste opazili, da se vaša datoteka `.emacs` sčasoma veča. To je vsekakor *Dobra Stvar*, ker pomeni, da ste našli način, kako prilagoditi Emacs, da bo deloval, kot **vi** želite. Prava sramota je, da tega ne omogoča večje število programske opreme.

Vsakič ko zaženete Emacs, le ta v vašem domačem imeniku poišče datoteko z imenom `.emacs`. Ta datoteka je mesto, kamor bi morali zapisati vso kodo v jeziku Lisp, za katero želite da se samodejno izvede.

Še en primer iz moje datoteke `.emacs`:

```
(setq inhibit-startup-message t)
```

Spremenljivka `inhibit-startup-message` nadzira ali bo Emacs ob zagonu izpisal pozdravno sporočilo. Čez nekaj časa sem se tega sporočila že pošteno naveličal, torej sem začel iskati način, kako bi ga izključil.

Za vajo poskusite sami tvoriti datoteko `.emacs` in vanjo dodajte to vrstico. Sedaj zaprite in ponovno zaženite Emacs. Pozdravnega sporočila ne bi smeli več videti.

Večkrat boste v dokumentaciji o načinih v urejevalniku Emacs (ali paketu) naleteli na predloge in primere, ki jih lahko vstavite v datoteko `.emacs`, da bi določen način ali paket deloval na poseben način.

Tudi GNU Emacs FAQ (C-h F) vsebuje nekaj tem, ki se nanašajo na datoteke `.emacs`. Mogoče vam bodo v pomoč.

4.3 Prilagoditveni paket

Čeprav je urejevalniku Emacs priljubljenost rasla, nadaljeval pa se je tudi njegov razvoj, je nekdo rekel “obstajati mora boljši način s katerim bodo lahko novi uporabniki prilagodili svoj urejevalnik Emacs.” Plod tega je prilagoditev.

Prilagoditev nudi več intuitivnih metod za prilagajanje delov urejevalnika Emacs. Da bi to preizkusili, pogledjte v podmeni `Customize` v vašem meniju `Help` ali pa vtipkajte `M-x customize`.

Prilagoditev deli prilagoditve v logične skupine, kot na primer “urejanje” (`Editing`), “programiranje” (`Programming`), “datoteke” (`Files`), itd. Nekatere skupine vsebujejo podskupine.

Če boste spremembe izvajali s pomočjo vmesnika za prilagoditve, jih bo Emacs shranil v vašo datoteko `.emacs`. To je res uporabno, ker lahko na ta način enostavno pregledate (in popravite) spremembe, ki jih bo opravil za vas.

Vmesnika za prilagoditve ne uporabljam, zaradi tega vam tudi ne morem dati več informacij o njem.

4.4 Prikaz v X Windows

kot vsaka aplikacija, ki se lepo obnaša v sistemu X, tudi Emacs upošteva vaše vire X. To pomeni, da lahko nadzirate barve, geometrijo in ostale stvari, ki so specifične za sistem X, prav tako, kot lahko to počnete s programom `xterm`, `nxterm`, oz. katerim koli.

Pa pogledjmo ustrezen del moje datoteke `~/.Xdefaults`:

```
emacs*Background: DarkSlateGray
emacs*Foreground: Wheat
emacs*pointerColor: Orchid
emacs*cursorColor: Orchid
emacs*bitmapIcon: on
emacs*font: fixed
emacs.geometry: 80x25
```

Več podrobnosti o virih sistema X boste našli na vaših straneh priročnika X.

Chris Gray (cgray4@po-box.mcgill.ca) je napisal še opombo:

V distribuciji Debian izgleda, da se datotek `~/.Xdefaults` ne uporablja. Uporabniki distribucije Debian lahko can put what you have given in `/etc/X11/Xresources/emacs` and they can have the pretty colors that they had when they were using RedHat.

5 Priljubljeni paketi

Kot dodatek k velikemu številu različnih načinov, ki so na voljo za urejevalnik Emacs, obstaja tudi veliko dodatnih (add-on) **paketov**. Jaz jih imenujem paketi zato, ker so več kot samo načini. Pogosto vsebujejo dodatne lastnosti ali

pa so tako veliki, da ne bi bilo pravično, če bi jih poimenovali z načini. V drugih primerih se uporabljajo tudi kot programska oprema, ki dopolnjuje ali vpeljuje druge načine ali pakete v urejevalnik Emacs. Razlika ni jasno začrtana.

5.1 VM (Pošta)

Če citiram dokument Odgovori na pogosto zastavljena vprašanja (FAQ) VM:

VM (View Mail - Ogled pošte) je podsistem urejevalnika Emacs, ki omogoča branje in odstranjevanje pošte kar v urejevalniku samem. Obstajajo ukazi, ki služijo za izvajanje operacij, običajnih za mail user agent, kot na primer tvorjenje odgovorov, shranjevanje sporočil v datoteke, brisanje sporočil, itd. Obstajajo tudi drugi naprednejši ukazi, ki omogočajo bursting and creating digests, message forwarding, and organizing message presentation according to various criteria.

Ko sem pričel uporabljati Emacs, sem preizkusil tudi VM. Ugotovil sem, da je odlično nadomestilo za programe Pine, Elm oz. večino vseh ostalih programov za pošto. Nisem pa želel uporabljati dveh različnih programov za branje pošte in novic. Danes se VM aktivno razvija in je dobro podprt.

Na voljo je na naslovu: <http://www.wonderworks.com/vm/>.

5.2 Gnus (pošta in novice)

Citat iz priročnika GNUS:

Gnus je laboratorij za branje sporočil. Vse stvari so urejene v novičarske supine. Z njim lahko berete pošto, brskate po imenikih, uporabljate ftp storitve — z njim lahko berete celo novice!

Gnus tries to empower people who read news the same way Emacs empowers people who edit text. Gnus ne postavlja omejitev o tem, kaj lahko uporabnik stori, ampak spodbuja ljudi, da razširijo Gnus tako, da se bo obnašal kot oni to želijo. Program ne bi smel nadzirati ljudi; ljudje bi morali imeti možnost, da počnejo kar jih je volja pri uporabi (oz. zlorabi) programa.

Tudi jaz trenutno za branje pošte in novic uporabljam GNUS (kot je bilo že zgoraj nakazano). Tudi GNUS se danes aktivno razvija in je dobro podprt.

Dobite ga lahko na naslovu: <http://www.gnus.org/>.

5.3 BBDB (rollodex)

BBDB je Insidious Big Brother Database, podoben programu rollodex za Emacs, ki deluje z večino priljubljenih paketov za pošto v urejevalniku Emacs (vključno z VM in GNUS).

Najdete ga na naslovu: <http://pweb.netcom.com/~simmonmt/bbdb/index.html>.

5.4 AucTeX (še en način TeX)

AucTeX je še eden od načinov za urejanje datotek tipa TeX.

Če citiram spletno stran AucTeX:

AUC TeX je prožen paket, ki podpira pisanje in oblikovanje datotek tipa TeX za večino različic urejevalnika GNU Emacs. Podprte so številne makro strani, vključno z AMS TeX, LaTeX in TeXinfo.

Najdete ga na naslovu: <http://sunsite.auc.dk/auctex/>.

6 Ostali viri

Ta razdelek opisuje knjige, spletne strani, novinarske skupine, poštne sezname in ostala mesta, kjer lahko najdete več informacij o urejevalniku Emacs.

6.1 Knjige

Za učenje urejevalnika Emacs obstaja nekaj res odličnih knjig. Ugotovili boste, da tudi veliko knjig za sisteme Unix in Linux vsebuje vsaj poglavje ali dva, ki opisuje urejevalnik Emacs (in `vi`).

6.1.1 Učenje uporabe urejevalnika GNU Emacs

Avtorji: Debra Cameron, Bill Rosenblatt, Eric S. Raymond

Založnik: O'Reilly & Associates - <http://www.ora.com/>

S popustom pa jo lahko kupite tudi na Amazon.com preko naslova: <http://www.amazon.com/exec/obidos/ASIN/1565921526/>

Komentar: Verjetno je najbolje začeti s to knjigo. Ko boste prebrali dokument HOWTO in se prebili skozi FAQ, vam bo ta knjiga služila kot razumljiva in zelo lahko dostopna vadnica.

6.1.2 Pisanje razširitev za urejevalnik GNU Emacs

Avtor: Bob Glickstein

Založnik: O'Reilly & Associates - <http://www.ora.com/>

S popustom pa jo lahko kupite tudi na Amazon.com preko naslova: <http://www.amazon.com/exec/obidos/ASIN/1565922611/>

Komentar: Če uporabljate Emacs že nekaj časa in ste se odločili, da bi radi napisali svoj lasten način ali pa bi radi samo preizkusili nekatere napredne prilagoditve, potem je to prava knjiga za vas. Ker vas ne namerava naučiti programiranja v jeziku Lisp, vsebuje samo kratek uvod v ta programski jezik.

6.1.3 Programiranje v Lispu: Uvod

Avtor: Robert J. Chassell

Iz datoteke README:

To je elementaren uvod v programiranje v jeziku Lisp za Emacs in je namenjen ljudem, ki niso programerji in jih programiranje tudi ne zanima, bi pa radi prilagodili in razširili svoje računalniško okolje.

Cel priročnik lahko najdete na strežniku FTP projekta GNU: <ftp://prep.ai.mit.edu/gnu/emacs/>.

S popustom pa jo lahko kupite tudi na Amazon.com preko naslova: <http://www.amazon.com/exec/obidos/ASIN/1882114418/jeremya>

Komentar: Knjiga predstavlja dober uvodni priročnik za jezik Lisp v urejevalniku Emacs -tudi če niste programer.

6.1.4 Referenčni priročnik za programski jezik Lisp v urejevalniku GNU Emacs

Avtor: Richard Stallman

Založnik: The Free Software Foundation - <http://www.fsf.org/>

Priročnik lahko v celoti najdete na strežniku GNU FTP: <ftp://prep.ai.mit.edu/gnu/emacs/>.

Komentar: To je najpomembnejši vodič v programski jezik Lisp za Emacs.

6.2 Spletne strani

6.2.1 EMACSulation

EMACSulation je članek, za katerega skrbi Eric Marsden in izhaja v reviji na zvezi Linux Gazette. Revijo lahko najdete na naslovu <http://www.linuxgazette.com/>. Zadnji članek v času pisanja tega dokumenta je na naslovu <http://www.linuxgazette.com/issue39/marsden.html>. Povezave na prejšnje članke pa boste našli na dnu dokumenta.

6.3 Novičarske skupine

Poiščite novičarske skupine, ki vsebujejo niz "emacs". Verjetno jih boste našli kar nekaj. Skupine, ki jih vsebuje moj strežnik so:

- comp.emacs
- comp.emacs.sources
- gnu.emacs
- gnu.emacs.bug
- gnu.emacs.help
- gnu.emacs.sources

6.4 Poštni seznam

Edini poštni seznam, ki je posvečen urejevalniku Emacs in ga v tem trenutku poznam je seznam NT-Emacs. To je seznam, namenjen ljudem, ki uporabljajo različico urejevalnika Emacs za Microsoft Windows. Več informacij boste našli v dokumentu NT-Emacs FAQ <http://www.cs.washington.edu/homes/voelker/ntemacs.html>.

6.5 Arhiv jezika Lisp za Emacs

Iz arhiva jezika Lisp za Emacs README:

Arhiv jezika Lisp za Emacs na [ftp.cis.ohio-state.edu](ftp://ftp.cis.ohio-state.edu) vsebuje številne koščke in pakete kode jezika Lisp za Emacs. Lisp za Emacs je jezik, ki se uporablja za razširjanje urejevalnika GNU Emacs, ki ga je izdala Free Software Foundation. Čeprav je v distribuciji urejevalnika GNU Emacs vključeno veliko kode jezika Lisp, je veliko ljudi napisalo pakete za komunikacijo z drugimi sistemi, za boljšo podporo urejanja programskega jezika, za dodajanje novih lastnosti ali za spreminjanje privzetega obnašanja urejevalnika Emacs. Večji del vsebine tega dokumenta so napisali posamezniki in jo javno razširili preko Interneta po poštnih seznamih info-emacs ali info-gnu-emacs ali po novičarskih skupinah comp.emacs, gnu.emacs ali gnu.emacs.sources.

Arhivi so na voljo na naslovu <ftp://ftp.cis.ohio-state.edu/pub/emacs-lisp/>.

NOTE: Moje mnenje je, da arhiv jezika Lisp za Emacs postaja že nekoliko zastarel. Objavi se zelo malo novih (ali posodobljenih) paketov, čeprav obstajajo. *Objavljeni* pa so na novičarski skupini comp.emacs.sources. (Če mislite, da se motim, me lahko popravite.)

7 Zasluge

K uspehu tega dokumenta so pripomogli naslednji ljudje.

- Robert Vollmert *rvollmer@gmx.net*
- Larry Brasfield *larrybr@seanet.com*
- Etienne Grossmann *etienne@anonimo.isr.ist.utl.pt*
- Thomas Weinell *kf6mli@amsat.org*
- Adam C. Finnefrock *adam@bigbro.biophys.cornell.edu*
- Chris Gray *cgray4@po-box.mcgill.ca*
- Robert J. Chassell *bob@rattlesnake.com*
- Isaac To *kkto@csis.hku.hk*
- Matteo Valsasna *valsasna@elet.polimi.it*
- Tijs van Bakel *smoke@casema.net*