LinuxDoc+Emacs+Ispell-HOWTO

Avtor: Philippe MARTIN (*feloy@wanadoo.fr*)
Prevajalec: Sébastien Blondeel (*Sebastien.Blondeel@lifl.fr*)
Prevajalca v slovenščino: Roman Maurer (prva 4 poglavja) in Jernej Kovačič (*jkovacic@email.si*) (ostalo) v0.5, 28. april 1998, prevod 10. september 2001

Ta sestavek je namenjen prevajalcem in piscem spisov HOWTO za Linux ali katerikoli drugih dokumentov Dokumentacijskega projekta za Linux. Ponuja jim namige za uporabo orodij, vključno z urejevalnikom besedil Emacs in črkovalnikom Ispell.

Kazalo

1 Predgovor

1.1 Pravice razširjanja

Copyright Philippe Martin 1998
Ta sestavek lahko razširjate in/ali spreminjate, če upoštevate pogoje GNU-jevske splošne licence različice 2 ali poznejše.

1.2 Zahvale

Posebej se zahvaljujem Sébastienu Blondeelu, ker me je tako mnogo vpraševal o nastavitvah Emacsa. Njegova bistra vprašanja so mi omogočila, da bolje poznam ta urejevalnik in s tem dokumentom prenesem znanje tudi na vas.

1.3 Pripombe

Ne oklevajte in mi sporočite vsako pripombo in/ali kritiko tega dokumenta, za katero mislite, da bo prispevala k izboljšavi.
Tudi ne oklevajte z zastavljanjem vprašanj, ki se nanašajo na tukaj obravnavane teme, z veseljem vam bom odgovoril.

1.4 Različice

Ta sestavek obravnava naslednje različice:

- SGML-Tools različice 0.99,
- Emacs različice 19.34,
- Ispell različice 3.1,
- Vse knjižnice Emacs, na katere se sklicuje ta sestavek, se razširjajo z gornjo različico Emacs, razen `iso-sgml`, ki pride poleg XEmacsa, in `psgml`, ki se razširja posebej.

2 Uvod

2.1 Jezik SGML

Jezik **SGML** (ang. *Standard Generalised Mark-up Language*) je jezik za definiranje tipov dokumentov.

Na primer, definira se lahko tip dokumenta za *kuharski recept*, katerega prvi del določa sestavine, drugi del pripomočke, tretji navodila za peko torte, v zadnjem delu pa je prikazana slika končnega rezultata.

Takšna definicija tipa dokumenta se imenuje DTD (ang. *Document Type Definition*). DTD ne določa izgleda končnega izdelka, temveč le definira, kaj vse lahko vsebuje.

Če spet uporabimo prejšnji primer: prepričan sem, da ste se po branju moje zamisli kuharskega recepta domislili svojega ali recepta vašega priljubljenega kuharja. Vseeno izgledajo drugačni: moj ima fotografijo, ki jo najdete v levem zgornjem kotu kopališke omarice, seznam sestavin pa najdete med bazenom in žarom. In vaš?

Zahvaljujoč tej standardni definiciji se lahko pišejo sestavki, ne da bi morali predvideti njihov izgled, kot ga bo videl bralec.

2.2 Definicija tipa `LinuxDoc`

Ta tip omogoča pisanje, kot ste verjetno uganili, dokumentacije, povezane z Linuxom.

Ti dokumenti so navadno sestavljeni takole: začenjajo se z naslovom, ki mu sledi avtorjevo ime in številka različice ter datum. Potem pride povzetek, ki kratko razlaga vsebino sestavka (da vam ni treba brskati po njej za ugotovitev, da to ni to, kar iščete), potem kazalo, ki prikazuje strukturo sestavka, ter omogoča tistim, ki se jim mudi, da preidejo neposredno na del, ki ga želijo brati.

In nadalje sledi zbirka poglavij, razdelkov, odstavkov. Med temi odstavki lahko vstavite kodo programov, spremenite pisavo, da ponazorite besedo ali stavek, vstavljate sezname, se sklicujete na druge dele sestavka, itd.

Za pisanje takšnega sestavka zadostuje na primernem mestu določiti naslov, avtorja, datum in različico sestavka, nato pisati poglavja in razdelke, povedati, kdaj se vstavi seznam in kaj so njegovi elementi, itd.

2.3 Paket `SGML-Tools`

Paket **SGML-Tools** omogoča izdelavo končnega formata dokumenta iz njegovega opisa. Če želite dokument uvrstiti v vašo osebno knjižnico, boste izbrali izhodni format *PostScript*, če ga želite deliti s svetom na svetovnem spletu, boste izbrali *HTML*, in če ne morete pomagati in morate stvar brati pod Windows, ga boste spremenili v *RTF*, da ga boste lahko uvozili v kakšen urejevalnik besedil. Morda boste želeli uporabljati različne fomrate, ki se bodo prilagajali vašem spremenljivem razpoloženju.

Paket SGML-Tools je dostopen po anonimnem FTP-ju z arhiva `<ftp://ftp.lip6.fr/pub/sgml-tools/>`.

3 Vaš prvi sestavek

3.1 Iz tekstovne datoteke

Če želite spremeniti vaš tekstovni dokument v DTD LinuxDoc zaradi spremembe v razne druge formate, je to prava pot:

1. Na samem začetku sestavka dodajte naslednje vrstice:

   ```
   <!doctype linuxdoc system>
   <article>
   <title>Tukaj je naslov</title>
   <author>
   ime avtorja, njegov e-poštni naslov
   </author>
   <date>
   ```

```
različica in datum sestavka
</date>
```

2. Če na začetu na kratko opišete vsebino sestavka, obkrožite ta odstavek z oznakamaa `<abstract>` in `</abstract>`.

3. Potem vstavite oznako `<toc>`, ki v končnem formatu samodejno naredi *kazalo* (ang. Table Of Contents).

4. Na začetku vsakega novega poglavja nadomestite vrstico s številko in naslovom poglavja z vrstico:

   ```
   <sect>Naslov poglavja
   ```

   in na koncu poglavja dodajte oznako `</sect>`.
   Vedite: Ni vam treba vstavljati številke poglavja, to se naredi samodejno.

5. Na enak način nadaljujte z razdelki. Morali boste odstraniti njihove številke in označiti njihov obseg z oznakama `<sect1>` in `</sect1>`.

6. Na podoben način lahko definirate do 4 nivoje gnezdenja razdelkov (pri `<sect4>` in `</sect4>`).

7. Na začetku vsakega odstavka vstavite oznako `<p>`.

8. Če želite poudariti določene dele ali fraze, jih vkleščite med oznaki `<it>` in `</it>` (*kurzivno*), ali `<bf>` in `</bf>` (**polkrepko**), ali med `<tt>` in `</tt>` (`pisalni stroj`).

9. Seznam, kot je naslednji:

   ```
   To je seznam s štirimi vrsticami:

    - tukaj je prva vrstica
    - potem druga
    - še ena
    - dovolj.
   ```

   vstavite z naslednjo kodo:

   ```
   To je seznam s štirimi vrsticami:
   <itemize>
   <item>tukaj je prva vrstica
   <item>potem druga
   <item>še ena
   <item>dovolj.
   </itemize>
   ```

10. Ko celoten blok predstavlja del programa ali kaj drugega, kar mora ohraniti svoj izgled:

    ```
    <verb>
    10 REM O, Bog, kaj pa je zdaj to?
    20 REM Mislil sem, da tega že zdavnaj ni več!
    30 PRINT "Vrnil sem se, ";
    40 PRINT "da bi rešil svet."
    50 INPUT "Pred kom, vprašujete? ",M$
    60 IF M$="Bill" THEN PRINT "Pametni ste.":GOTO VRAJ
    70 ELSE PRINT "Ničesar ne razumete...":GOTO DAJDOLARJE
    </verb>
    ```

11. Prišli ste do točke, ko so vaše formatirne zmogljivosti za SGML DTD LinuxDoc precej spodobne. Če želite izboljšati svoj sestavek, poglejte v uporabniški priročnik paketa **SGML-Tools**, ki podrobneje razlaga tip dokumenta **LinuxDoc**.

4 Prilagoditev Emacsa

4.1 Šumniki in črke z akcenti

Črke slovenske abecede so podprte v kodnem naboru ISO Latin 2 (ISO 8859-2). Na tem mestu je koristno prebrati *Slovenian HOWTO*, ki razlaga nastavitve za vnos, prikaz in tiskanje naših črk.

Emacs prebere svoje nastavitve iz naslednjih datotek:

- `/usr/lib/emacs/site-lisp/site-start.el` (skupne nastavitve)
- `~/.emacs` (osebne nastavitve)
- `/usr/lib/emacs/site-lisp/default.el` (skupne nastavitve)

Kjer se sklicujemo na datoteko `.emacs`, pravzaprav s tem mislimo na katerokoli izmed zgornjih treh datotek.

4.1.1 Prikaz 8-bitnih znakov

Če želite pisati spise `.sgml` v slovenščini ali v kateremkoli drugem evropskem jeziku, boste potrebovali 8-bitne znake. Emacs prilagodite za prikaz 8-bitnih znakov z naslednjo vrstico v vaši datoteki `.emacs`:

```
(standard-display-european t)
```

Za vnos zahodnoevropskih črk pride prav tudi naslednja vrstica, ki pa zaenkrat ni uporabna za slovenščino:

```
(load-library "iso-syntax")
```

Če uporabljate Emacs za pisanje zahodnoevropskih besedil na terminalu, ki ne podpira 8-bitnega prikaza, lahko uporabite knjižnico `iso-ascii` (`(load-library "iso-ascii")`), ki poskrbi, da Emacs prikazuje najboljše približke 8-bitnih črk.

4.1.2 Vnos 8-bitnih znakov

Tipkovnico nastavite za vnos 8-bitnih črk v konzoli ali pod okni X, če Emacs poganjate tam. Tovrstne nastavitve so skupne vsem programom pod konzolo ali okni X in so opisane v spisu *Slovenian HOWTO*.

Če vaša tipkovnica v konzoli ali pod oknih X torej podpira vnos 8-bitnih znakov, boste morali le v datoteko `.emacs` dodati naslednje vrstice:

```
;;  Omogočimo vnos znakov s kodo večjo od 127
(set-input-mode (car (current-input-mode))
        (nth 1 (current-input-mode))
        0)
```

Če vaš terminal ne podpira 8-bitnega vnosa, ste pri pisanju slovenskih besedil v težavah. Za pisanje zahodnoevropskih besedil si lahko pomagate z dodatnimi paketi:

Knjižnica `iso-acc` Knjižnica za Emacs `iso-acc` vam bo omogočila vnos 8-bitnih znakov s tipkovnico, ki sicer omogoča le vnos 7-bitnih črk.

Za uporabo te knjižnice vstavite naslednje v vašo datoteko `.emacs`:

```
(load-library "iso-acc")
```

Ko boste naslednjič pognali Emacs in odprli datoteko, ki bi jo radi uredili, napišite `Meta-x iso-accents-mode`.

Črko **é** v francoski besedi *café* lahko vstavite tako, da pritisnete **'** in potem še `e`. V splošnem lahko vpišete zahodnoevropski znak z akcentom tako, da najprej pritisnete ustrezen akcent, ki mu sledi črka pod njim (velika ali mala). Uporabite lahko naslednje akcente:

- ' : ostrivec
- ` : krativec
- ˆ : strešica
- ¨: preglas
- ˜ : tilda nad črko, sedij (kljukica pod „c“), in drugi posebni primeri (glejte `iso-acc.el`)
- / : različni skandinavski akcenti, idr.

Če potrebujete enega od teh znakov in ne naglašene črke, pritisnite za njim presledek. Na primer, za vnos *l'éléphant*, napišite `l '` ⟨*spc*⟩ `' e l ' e ...`

Vse možne kombinacije boste našli v datoteki `iso-acc.el`.

Tipka ⟨*Meta*⟩ Nekateri terminali vam omogočajo vnos 8-bitnih znakov s pomočjo tipke ⟨*Meta*⟩ (ali ⟨*Alt*⟩). Na primer, če pritisnete kombinacijo ⟨*Meta*⟩**-i** lahko dobite znak **é**.

Vendar Emacs potrebuje tipko ⟨*Meta*⟩ za drugačno uporabo. Zato morate pritisniti zaporedje ⟨*Ctrl*⟩-q (ukaz `quoted-insert`) preden pritisnete ⟨*Meta*⟩-i.

Ta ukaz vam dovoljuje vnos znakov tudi z vpisom osmiške kode. Pritisnite kombinacijo ⟨*Meta*⟩-q, ki ji sledi osmiška koda znakov, ki bi jih radi vnesli. Slovenske šumnike „ČŠŽ“ in hrvaški črki „ĆĐ“, pisane po ISO Latin 2, lahko vnesete z naslednjimi kombinacijami:

```
č  Ctrl-q 350    Č  Ctrl-q 310
š  Ctrl-q 271    Š  Ctrl-q 251
ž  Ctrl-q 276    Ž  Ctrl-q 256
ć  Ctrl-q 346    Ć  Ctrl-q 306
đ  Ctrl-q 360    Đ  Ctrl-q 320
```

4.1.3 Prikaz 8-bitnih znakov v SGML DTD LinuxDoc

V SGML lahko vnašate naglašene znake z makroukazi, vendar žal ne tudi srednjeevropskih. Na primer, simbol za tipko **é** je **é**. V splošnem, aplikacije, ki morajo brati SGML znajo brati 8-bitne znake in ni potrebno uporabljati teh makroukazov. Vendar nekateri morda ne bodo mogli brati 8-bitnih znakov. Če vemo, da obstaja preprosta rešitev (spet samo za zahodne narode, op. prev.), bi bilo škoda pustiti, da se takšne aplikacije sesujejo.

Knjižnica `iso-sgml` vam omogoča vnos naglašenih znakov v Emacsu, kot vedno, a ob shranjevanju vaše datoteke na disk, bo samodejno spremenila 8-bitne znake v njihove ekvivalente v SGML. Vendar, žal, slovenske povsem napačno, saj jih v ISO Latin 1 ni.

No, zahvaljujoč tej knjižnici je lahko vnašati in brati spise z zahodnimi znaki, pisanimi po ISO Latin 1 (ISO 8859-1) v Emacsu, in prepričani ste lahko, da bo aplikacija, ki sicer ne sprejme 8-bitnih sestavkov, po takem postopku vašega sprejela.

Za uporabo te knjižnice morate dodati naslednje vrstice v vašo datoteko `.emacs`:

```
(setq sgml-mode-hook
'(lambda () "Privzete nastavitve načina SGML."
(load-library "iso-sgml")))
```

4.2 Način SGML

Po nalaganju datoteke s podaljškom **.sgml**, Emacs samodejno vstopi v **način sgml** (ang. sgml mode). Če ne, mu lahko to naročite ročno z ukazom `Meta-x sgml-mode`, ali samodejno z dodatkom naslednjih vrstic v vašo datoteko `.emacs`:

```
(setq auto-mode-alist
(append '(("\.sgml$"  . sgml-mode))
        auto-mode-alist))
```

Ta način vam bo, na primer, omogočil izbiro tega, kako boste vnašali 8-bitne znake. Z ukazom `Meta-x sgml-name-8bit-mode` (ali z menujsko postavko *SGML/Toggle 8-bit insertion*), lahko izberete vnos 8-bitnih znakov kot takih, ali v obliki SGML, se pravi, v obliki **&...;**.

Emacs vam bo tudi dovoljeval skriti ali prikazati oznake SGML, z ukazom `Meta-x sgml-tags-invisible` (ali z menujsko postavko *SGML/Toggle Tag Visibility*).

4.3 Način PSGML

Načinu PSGML zelo pomaga pri urejanju spisov SGML z Emacsom.

Dokumentacija za *psgml-linuxdoc* razlaga, kako namestiti in uporabljati ta način z zvrstjo besedila *LinuxDoc*.

4.4 Razno

4.4.1 Način auto-fill

Ko v normalnem načinu napišete odstavek in pridete do konca vrstice, morate sami uporabiti tipko ⟨*Return*⟩, da pridete v naslednjo vrstico, sicer se bo vaša vrstica nadaljevala nedoločeno dolgo skozi cel odstavek. Kadar uporabite ⟨*Return*⟩ za pomik v novo vrstico, dobite odstavek z neporavnanimi desnimi robovi. Če dopustite, da so nekatere daljše od razumne širine, jih z nekaterimi urejevalniki ne boste mogli videti.

Način **auto-fill** avtomatizira to nehvaležno opravilo: ko greste dlje od določenega stolpca (privzeto 70-tega), vas samodejno postavi v naslednjo vrstico.

Takole uporabite ta način in nastavite širino vaših vrstic na 80:

```
(setq sgml-mode-hook
     '(lambda () "Defaults for SGML mode."
         (auto-fill-mode)
         (setq fill-column 80)))
```

5 Ispell

Če želite pravopisno preveriti vaš spis v Emacsu, lahko uporabite paket **Ispell** in njegov način za Emacs.

5.1 Izbira privzetega besednjaka

Emacs lahko nastavite tako, da po nalaganju datoteke avtomatsko določi, kateri besednjak naj uporabi (uporabljate jih namreč lahko več). Prvi, zagotovo najpomembnejši, je glavni besednjak, ki se razširja skupaj z Ispellom. Izbirate lahko med več jeziki. Drugi besednjak je vaš osebni, kamor Ispell vstavlja besede, ki jih ni našel v prvem besednjaku, vendar ste mu naročili, naj si jih zapomni.

Če želite kot privzetega uporabljati slovenski besednjak (dobite ga lahko na naslovu *ftp://sizif.uni-lj.si/pub/i18n/ispell/ispell-sl-0.1-1.src.rpm*) in če želite uporabljati datoteko `.ispell-dico-perso` v vašem domačem imeniku kot osebni besednjak, vnesite naslednje vrstice v vašo datoteko `.emacs`:

```
(setq sgml-mode-hook
'(lambda () "Privzeto za nacin SGML."
(setq ispell-personal-dictionary "~/.ispell-dico-perso")
(ispell-change-dictionary "slovensko")
))
```

5.2 Izbira posebnih besednjakov za določene datoteke

Do manjšega problema lahko pride, če ne preverjate črkovanja v dokumentih vedno v istem jeziku. Če prevajate dokumente, je zelo verjetno, da pogosto pomešate jezike (in besednjake).

Žal ne poznam nobenega načina v Lispu, niti avtomatskega niti kakšnega z enojnim klikom z miško, za izbiro glavnih in osebnih besednjakov, povezanih s trenutno uporabljanim jezikom. (Če veste kaj več, mi prosim povejte!)

Vendarle pa je mogoče na koncu datoteke označiti, katere besednjake želite uporabljati ob preverjanju trenutne (in samo te) datoteke. Zadostuje, da te oznake dodate kot komentarje, tako da jih bo Ispell znal prebrati ob zagonu preverjanja črkovanja:

```
<!-- Local IspellDict: english -->
<!-- Local IspellPersDict: ~/emacs/.ispell-english -->
```

Če ste prej v datoteki `.emacs` definirali, da naj bodo privzeti besednjaki slovenski, lahko dodate te vrstice na konec vsake datoteke v angleščini.

5.3 Preverjanje črkovanja v vašem dokumentu

Preverjanje črkovanja celotnega dokumenta lahko s kateregakoli mesta v preverjanem dokumentu poženete z ukazom `Meta-x ispell-buffer`. Lahko pa tudi preverite samo del dokumenta:

- Označite začetek področja s tipkama `Ctrl-Spc` (mark-set-command),
- pojdite na konec področja, ki ga želite preveriti
- in vtipkajte `Meta-x ispell-region`.

Emacs zatem požene Ispell. Ko naleti na neznano besedo, vam jo pokaže (ponavadi osvetljeno) in čaka, da pritisnete na ustrezno tipko:

- **spc** sprejme besedo, samo tokrat,
- **i** sprejme besedo in jo doda v vaš osebni besednjak,
- **a** sprejme besedo, velja samo za to preverjanje,
- **A** sprejme besedo samo za to datoteko in jo doda v besednjak za preverjano datoteko,
- **r** dovoli vam ročno popravljanje besede

- **R** dovoli vam popravek ki velja za vse primere narobe vtipkane besede,
- **x** ustavi preverjanje in postavi kurzor nazaj,
- **X** ustavi preverjanje in pusti kurzor, kjer je, tako lahko popravite datoteko; s preverjanjem črkovanja lahko nadaljujete pozneje, če vtipkate `Meta-x ispell-continue`,
- **?** ponudi vam sprotno pomoč.

Če Ispell najde eno ali več besed, ki so podobne neznani, vam jih bo pokazal v majhnem okencu. Pred vsako stoji številka. Samo vtipkajte to številko in narobe vtipkana beseda se bo nadomestila z izbrano.

5.4 Osebni besednjak proti besednjaku za preverjano datoteko

Tipka **i** vam omogoča vnos besede v vaš osebni besednjak, **A** pa vam doda besedo v besednjak za preverjano datoteko.

Besednjak za preverjano datoteko je zaporedje besed, dodano na konec datoteke v obliki komentarjev. Ispell jih znova prebere vsakič, ko ga poženemo za preverjanje te datoteke. Na ta način lahko sprejmete nekaj besed, ki so sprejemljive samo v tej datoteki, v drugih pa morebiti ne.

Kar zadeva mene, mislim, da je bolje, da je osebni besednjak rezerviran za besede, ki niso vsebovane v glavnem besednjaku, vendar pripadajo jeziku (npr. besede z vezaji), in pa še za besede, kot so lastna imena idr. (npr. *Linux*), če niso podobne resničnim besedam iz glavnega besednjaka. Dodajanje preveč besed (npr. imen) v osebni besednjak je lahko nevarno, saj lahko izgledajo kot povsem običajna beseda v nekem jeziku.

5.5 Preverjanje črkovanja med tipkanjem

Ispell zna preverjati črkovanje v vaši datoteki kar med tipkanjem. Da bo to delovalo, morate uporabiti način **ispell-minor-mode**. Poženete ali ustavite ga, če vtipkate `Meta-x ispell-minor-mode`. Ispell bo *zapiskal* vsakič, ko vtipkate besedo, ki je ne prepozna.

Če vas ti *piski* motijo (ali če vaš sostanovalec že spi), lahko te moteče *piske* nadomestite z bliskanjem na zaslonu, kar dosežete z ukazom `Meta-x set-variable RET visible-bell RET t RET`. V datoteko `.emacs` lahko dodate naslednjo vrstico in za vedno utišate Emacs:

```
(setq visible-bell t)
```

6 Umazani triki

6.1 Avtomatsko dodajanje zaglavja...

Emacs vam omogoča nastaviti, da se nekatere akcije *sprožijo* ob določenih dogodkih (npr. odpiranje datoteke, shranjevanje, zagon novega načina itd.).

Knjižnica **autoinsert** uporablja naslednjo lastnost: ko v Emacsu odprete novo datoteko, vanjo ta knjižnica doda *standardno* (glede na tip datoteke) zaglavje.

V našem primeru bi lahko to *standardno* zaglavje določilo tip dokumenta (LinuxDoc), naslov, avtorja in datum.

Tukaj bom opisal dva načina za dodajanje takšnega zaglavja. Vrinete lahko datoteko s predlogo, ki vsebuje vse naštete informacije, lahko pa poženete rutino v **elispu**.

6.1.1 z vrivanjem datoteke

Najprej morate povedati Emacsu, naj ob odpiranju datoteke požene `auto-insert`, potem naj prebere knjižnico **autoinsert**, ki določi seznam `auto-insert-alist`, tega pa moramo popraviti. Ta seznam določa zaglavje, ki se bo vrinilo v odvisnosti glede na tip datoteke. Privzeto mora biti datoteka za vrivanje v imeniku `~/insert/`, vendar je mogoče popraviti vrednost spremenljivke `auto-insert-directory`, če želite imeti datoteko kje drugje.

V vašo datoteko `.emacs` dodajte naslednje vrstice, da se bo datoteka `~/emacs/sgml-insert.sgml` vrinila vsakič, ko odprete novo datoteko SGML:

```
(add-hook 'find-file-hooks 'auto-insert)
(load-library "autoinsert")
(setq auto-insert-directory "~/emacs/")
(setq auto-insert-alist
      (append '((sgml-mode .  "sgml-insert.sgml"))
              auto-insert-alist))
```

Zatem si lahko v datoteko `~/emacs/sgml-insert.sgml` napišete ustrezno zaglavje, znova poženete Emacs in odprete neko datoteko `foobar.sgml`. Emacs bi vam moral reči, da potrdite avtomatsko dodajanje zaglavja, in če odgovorite z 'yes', ga doda.

6.1.2 s funkcijo v elispu

To deluje podobno kot prej, le da morate namesto nastavljanja seznama `auto-insert-alist` nastaviti funkcijo, ki naj se izvrši. Prikazali bomo ta postopek, pri čemer bomo predpostavili, da želite naisati to funkcijo v datoteko `~/emacs/sgml-header.el`. (Nobene potrebe ni, da datoteko `.emacs` zasmetite s takšniimi funkcijami, ker se lahko zgodi, da bo predolga):

```
(add-hook 'find-file-hooks 'auto-insert)
(load-library "autoinsert")
(add-to-list 'load-path "~/emacs")
(load-library "sgml-header")
(setq auto-insert-alist
      (append '(((sgml-mode .  "SGML Mode") . insert-sgml-header))
              auto-insert-alist))
```

V A (dodatku) boste našli primer funkcije `insert-sgml-header`.

A Funkcija `insert-sgml-header`

Ta funkcija uporabniku omogoči dodajanje zaglavja za dokument dokumentacijskega projekta za Linux (Linux documentation Project) v datoteko. Izvede se lahko samodejno, ko nekdo odpre novo datoteko, ali pa izrecno na uporabnikovo željo.

Ta funkcija v *mini-bufferju* (spodnja vrstica v Emacsu) vpraša uporabnika za nekaj podatkov, izmed katerih so nekateri obvezni, nekateri pa ne.

Najprej se pojavi vprašanje o naslovu dokumenta. Če ne podate ničesar, se funkcija takoj konča, ne da bi se dodalo kakšno zaglavje. Sledijo vprašanja o datumu, avtorju, njegovem elektronskem naslovu in domači strani (zadnja dva podatka sta neobvezna).

Sledi zahteva po imenu prevajalca. Če ga ni, samo pritisnite *Enter*, in nobenih nadaljnjih vprašanj o morebitnem prevajalcu ne bo več. Če pa prevajalec obstaja, boste vprašani še po njegovem elektronskem naslovu in domači strani (oboje je ravno tako neobvezno).

Funkcija nato zapiše zahteve na mesto, kjer urejate pravkar odprto novo datoteko (buffer v Emacsovi terminologiji), seveda se dodajo vsi malo prej vtipkani podatki in pa še vse oznake, ki služijo označevanju povzetka (angl. abstract) in prvega poglavja. Končno se kurzor postavi na mesto, kjer je potrebno napisati povzetek.

```
(defun insert-sgml-header ()
  "Doda zaglavje v dokument LinuxDoc"
  (interactive)
  (let (title author email home translator email-translator home-translator date
             starting-point)
    (setq title (read-from-minibuffer "Naslov: "))
    (if (> (length title) 0)
        (progn
          (setq date (read-from-minibuffer "Datum: ")
                author (read-from-minibuffer "Avtor: ")
                email (read-from-minibuffer "Avtorjev elektronski naslov: ")
                home (read-from-minibuffer "Avtorjeva domaca stran: http://")
                translator (read-from-minibuffer "Prevajalec: "))
          (insert "<!doctype linuxdoc system>\n<article>\n<title>")
          (insert title)
          (insert "</title>\n<author>\nAuthor: ") (insert author) (insert "<newline>\n")
          (if (> (length email) 0)
              (progn
                (insert "<htmlurl url=\"mailto:")
                (insert email) (insert "\" name=\"") (insert email)
                (insert "\"><newline>\n")))
          (if (> (length home) 0)
              (progn
                (insert "<htmlurl url=\"http://")
                (insert home) (insert "\" name=\"") (insert home)
                (insert "\">\n<newline>")))
          (if (> (length translator) 0)
              (progn
                (setq email-translator (read-from-minibuffer "Prevajalcev el. naslov: ")
                      home-translator (read-from-minibuffer "Prevajalceva domaca stran: http://"))
                (insert "Translator : ")
                (insert translator)
                (insert "<newline>\n")
                (if (> (length email-translator) 0)
                    (progn
                      (insert "<htmlurl url=\"mailto:")
                      (insert email-translator) (insert "\" name=\"")
                      (insert email-translator)
                      (insert "\"><newline>\n")))
                (if (> (length home-translator) 0)
                    (progn
                      (insert "<htmlurl url=\"http://")
                      (insert home-translator) (insert "\" name=\"")
```

```
                (insert home-translator)
                (insert "\"><newline>\n")))))
(insert "</author>\n<date>\n")
(insert date)
(insert "\n</date>\n\n<abstract>\n")
(setq point-beginning (point))
(insert "\n</abstract>\n<toc>\n\n<sect>\n<p>\n\n\n</sect>\n\n</article>\n")
(goto-char point-beginning)
))))
```