LinuxDoc+Emacs+Ispell-HOWTO

Auteur: Philippe MARTIN (*feloy@wanadoo.fr*)
Vertaler: Sébastien Blondeel (*Sebastien.Blondeel@lifl.fr*) Vertaler (nl): Ellen Bokhorst (*bokkie@nl.linux.org*)
v0.4, 27 Februari 1998

Dit document is bedoeld voor schrijvers en vertalers van Linux HOWTO's of enig ander document voor het Linux Documentatie Project. Het geeft aanwijzingen voor het gebruiken van hulpmiddelen waaronder Emacs en Ispell.

Inhoudsopgave

1 Inleiding

1.1 Copyright

1.2 Verdiensten

Speciale dank gaat uit naar Sébastien Blondeel, een akelige vent die me nogal veel vroeg over het instellen van Emacs. Zijn slimme vragen lieten me het beter begrijpen en de kennis via dit document aan je door te geven.

1.3 Commentaar

Aarzel niet me te vertellen waarvan je denkt dat het kan helpen dit document te verbeteren. Ik zal je kritiek door en door bestuderen.
Aarzel ook niet om me vragen te stellen die te maken hebben met de onderwerpen die hier worden besproken. Ik zal ze met plezier beantwoorden, aangezien ze me kunnen helpen dit document verder te verbeteren.[1]

1.4 Versies

Dit document gaat uit van de volgende versies:

- Sgml-tools versie 0.99,
- Emacs versie 19.34,
- Ispell versie 3.1,
- Alle Emacs libraries waarnaar in dit document wordt verwezen, worden met de bovengenoemde versie van Emacs gedistribueerd, afgezien van `iso-sgml`, dat met XEmacs wordt gedistribueerd, en `psgml` is een stand-alone library.

[1] Opmerking van de vertaler: als de vertaling niet deugt, dan gaat dat naar mij!

2 Introductie

2.1 SGML

Standard Generalised Mark-up Language, of **SGML**, is een taal voor het definiéren van documenttypen.

Men kan het documenttype bijvoorbeeld definiëren als een *recept*, met een eerste deel dat de ingrediënten presenteert, een tweede deel dat de accesoires introduceert, een derde deel waarin stap-voor-stap instructies worden gegeven voor het bakken van de cake, en een aardig laatste plaatje om de uitkomst van dit alles te tonen.

Dit wordt een *Document Type Definition* genoemd. Het definieert niet hoe het uiteindelijke product er uit zal zien, het definieert slechts wat er in zou kunnen staan.

Om nogmaals hetzelfde voorbeeld aan te halen, ik ben er zeker van dat bij het lezen van mijn idee van een recept, je jouw favoriete bereidingen herkent. Niettemin zien ze er in feite anders uit: van de mijne is er een plaatje in de bovenste linkerhoek van de badkamerkast, en de lijst met ingrediënten bevindt zich in de achtertuin, tussen het zwembad en de barbecue. En de jouwe?

Met dank aan deze standaarddefinitie; men kan een document schrijven zonder acht te slaan op hoe het er uiteindelijk richting de gebruiker uit komt te zien.

2.2 De `LinuxDoc` Type Definitie

Dit type wordt gebruikt, zoals je misschien al hebt geraden, om documenten gerelateerd aan Linux, te schrijven.

Dergelijke documenten worden in het algemeen als volgt opgebouwd: ze beginnen met een titel gevolgd door de naam van de auteur, en het versienummer en de -datum. Vervolgens komt de samenvatting (zodat je er niet doorheen hoeft te bladeren, voordat je je achteraf realiseert dat het niet hetgeen is waarnaar je zocht), en dan de inhoud die de structuur toont, zodat degene die haast hebben, direct naar dat deel kunnen gaan wat ze willen lezen.

Vervolgens komt er een lijst met hoofdstukken, secties en paragrafen. Hiertussen kan men stukken programma invoegen, het font wijzigen om een woord of een zin te benadrukken, lijsten invoegen, naar een ander deel van het document verwijzen, enz.

Om een dergelijk document te schrijven, zul je gewoon op het juiste moment de titel, de auteur, de datum en de versie van het document, de hoofdstukken en paragrafen moeten schrijven en laten we zeggen, welke elementen er in een lijst staan als deze moeten worden ingevoegd, enz.

2.3 `SGML-Tools`

SGML-Tools zal de specificatie van een document in het uiteindelijke resultaat omzetten in de vorm waaraan je de voorkeur geeft. Als je wilt dat het een onderdeel van je persoonlijke bibliotheek vormt, zul je voor *PostScript* kiezen. Als je het wilt delen met de rest van de wereld via het Web, zal het *HTML* zijn. Als je er niets aan kunt doen en het onder Windows moet lezen, kun je het omzetten in *RTF* zodat je het in iedere tekstverwerker in kunt lezen. Of misschien wil je alle drie de formaten gebruiken om aan je wisselende stemmingen tegemoet te komen.

SGML-Tools zijn beschikbaar via anonieme FTP op *ftp://ftp.lip6.fr/pub/sgml-tools/*

3 Je eerste document

3.1 Vanuit een tekstdocument

Als je een tekstdocument in SGML wilt omzetten om het over te zetten naar andere formaten, is dit de manier waarop je 't doet:

1. Voeg de volgende regels aan het begin toe:

   ```
   <!doctype linuxdoc system>
   <article>
   <title>De titel staat hier</title>
   <author>
   naam van de auteur, e-mail van de auteur, enz.
   </author>
   <date>
   versie en datum
   <</date>
   ```

2. Als je aan het begin van het document de inhoud ervan in het kort beschrijft, omsluit die paragraaf dan met de `<abstract>` en `</abstract>` tags.

3. Voeg dan de `<toc>` tag in, wat staat voor *Inhoudsopgave*.

4. Aan het begin van ieder nieuw hoofdstuk, vervang je de regel waarin het nummer en de titel van het hoofdstuk staan door:

   ```
   <sect><heading>De Titel Van Het Hoofdstuk</heading>
   ```

   en voeg de `</sect>` tag aan het einde van het hoofdstuk toe.
   Opmerking: Je hoeft het hoofdstuknummer niet aan te geven, dit gebeurt automatisch.

5. Ga op dezelfde wijze door met paragrafen. Je zult de nummers moeten verwijderen en de titels laten beginnen met de `<sect1>` tag en laten beëindigen met de `</sect1>` tag.

6. Je kunt op vergelijkbare wijze 4 niveuas in de secties nesten, door gebruik te maken van `<sectn>` en `</sectn>` waar de `n=` staat voor 2, 3, of 4.

7. Aan het begin van iedere paragraaf voeg je de `<p>` tag in.

8. Als je een aantal delen wilt benadrukken, omsluit ze dan met de `<it>` en `</it>` (*cursief*), `<bf>` en `</bf>` (**vet**), of `<tt>` en `</tt>` (`typewriter stijl`).

9. Om een op de volgende gelijkende lijst in te voegen:

   ```
     Dit is een lijst met vier regels:

   - hier komt de eerste regel
   - vervolgens komt de tweede regel
   - nog een andere
   - dat is het.
   ```

je moet dit vervangen door:

```
Dit is een lijst met vier regels:
<itemize>
<item>hier komt de eerste regel</item>
<item>vervolgens komt de tweede regel</item>
<item>nog een andere</item>
<item>dat is het.</item>
</itemize>
```

10. Als een geheel blok uit een gedeelte van een programma bestaat, of iets anders dat in 't oog moet springen:

```
10 REM Oh mijn God wat is dit?
20 REM Ik dacht dat dit al lang niet meer bestond!
30 PRINT "Ik ben terug bij";
40 PRINT "red de wereld."
50 INPUT "Van wie denk je dat 't is? ",M$
60 IF M$="Bill" THEN PRINT "Gij bent verstandig.":GOTO PARADISE
70 ELSE PRINT "Je snapt er niks van...":GOTO RICHMOND
</verb>
```

11. Tot dusverre zijn je SGML-formatterings vaardigheden al tamelijk goed. Als je je document bij wilt schaven, zou je de gebruikersgids **SGML-Tools** eens kunnen bekijken, waarin meer details staan over het **LinuxDoc** documenttype.

4 Configureren van Emacs

4.1 Geaccentueerde Tekens

Als je documenten in het Frans of elke andere West-Europese taal wilt schrijven, zul je 8-bit tekens nodig hebben. Je kunt Emacs als volgt instellen dat het dergelijke tekens accepteert.

4.1.1 Het tonen van 8-bit tekens

Om in Emacs 8-bits tekens te kunnen tonen, zul je de volgende regels in het bestand `.emacs` nodig hebben:

```
(standard-display-european 1)
(load-library "iso-syntax")
```

Als je Emacs op een terminal gebruikt die geen ondersteuning voor 8-bit geeft, kun je de `iso-ascii` library (`(load-library ïso-ascii")`) gebruiken, waarmee Emacs de opdracht krijgt om dergelijke tekens zo goed mogelijk te benaderen.

4.1.2 Het typen van 8-bit tekens

Als je toetsenbord je de mogelijkheid geeft om geaccentueerde tekens in te voeren, geen probleem. Als dit niet zo is zijn hier een aantal remedies:

De iso-acc-library De Emacs `iso-acc` library geeft je de mogelijkheid om 8-bits tekens vanaf een 7-bit toetsenbord in te tikken.

Om het te kunnen gebruiken, voeg je de volgende regel toe aan het bestand `.emacs`:

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

Vervolgens, als je Emacs draait en het bestand opent dat je wilt wijzigen, tik je in: `Meta-x iso-accents-mode`.

Je kunt dan de **é** van het Franse woord *café* invoegen door **'** te typen en vervolgens `e`. Meer in het algemeen, je zult een geaccentueerd teken intikken door als eerste het accent in te tikken en vervolgens de letter die moet worden geaccentueerd (kleine- of hoofdletter). Je kunt de volgende accenten gebruiken:

- **'** : Accent Aigu
- **`** : Accent Grave
- **^** : Accent Circonflexe
- "

  : Dubbel aanhalingsteken
- **~** : Tilde, cedille, en andere speciale gevallen (cf iso-acc.el).
- / : Om tekens te scheiden, enz.

Als je één van deze tekens nodig hebt en niet een geaccentueerde letter, typ je er een spatie achter. Om bijvoorbeeld *l'éléphant* te typen, typ je `l ' ⟨spc⟩ ' e l ' e ...`

Alle mogelijke combinaties zijn in het bestand `iso-acc.el` te vinden.

De Meta-toets Een aantal terminals staan je toe om 8-bits tekens in te tikken met de <Meta> (of <Alt>) toets. Als je bijvoorbeeld **<Meta>-i** indrukt, zul je het **é**-teken krijgen.

Maar Emacs heeft de <Meta>-toets voor andere doeleinden gereserveerd, en ik ken geen library die je het laat gebruiken voor geaccentueerde tekens.

Dit is een oplossing:

```
(global-set-key "\ei" '(lambda () (interactive) (insert ?\351)))
   -                                              ---
```

Als je een dergelijk regel in het bestand `.emacs` invoegt, kun je een **é** intikken door gebruik te maken van de **<Meta>-i** toetsaanslag. Je kunt op die manier de combinaties, die je nodig hebt, herdefiniëren als je de **i** door de juiste toets en **351** door de juiste code (de code uit de ISO-8859-1 tekenset) vervangt.

Waarschuwing! Het kan zijn dat een aantal lokale modes dergelijke toetscombinaties herdefiniëren.

4.1.3 Het tonen van 8-bits SGML-tekens

Onder SGML kun je geaccentueerde tekens met macros intikken. De **é** toets bijvoorbeeld is **é**. In het algemeen kunnen de applicaties die SGML in moeten kunnen lezen, 8-bit tekens inlezen en is het dus niet nodig om deze macro's te gebruiken. Maar een aantal ervan kunnen dit mogelijk niet. Aangezien er een manier is om dit probleem op te lossen, zou het zonde zijn om het hierdoor vast te laten lopen.

De `iso-sgml` library laat je zoals altijd geaccentueerde tekens onder Emacs typen, maar bij het opslaan van je bestand naar disk, zal het deze 8-bits tekens omzetten in de SGML-equivalente.

Dankzij deze library is het daarom makkelijk om je document onder Emacs in te typen en het opnieuw in te lezen, zodat je er zeker van kunt zijn dat een niet-8-bit-clean-applicatie je document zal accepteren.

Om deze bibliotheek te kunnen gebruiken, hoef je slechts de volgende regels aan het bestand `.emacs` toe te voegen:

```
(setq sgml-mode-hook
'(lambda () "Standaards voor SGML mode."
 (load-library "iso-sgml")))
```

4.2 SGML mode

Bij het laden van een bestand met de **.sgml** extensie, schakelt Emacs automatisch over op de **sgml mode**. Als dit niet gebeurt, kun je dit handmatig doen door `Meta-x sgml-mode` in te tikken, of automatisch door de volgende regels aan het `.emacs` bestand toe te voegen:

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

Deze mode laat je kiezen uit hoe je bijvoorbeeld 8-bit tekens in wilt voegen. Met de `Meta-x sgml-name-8bit-mode` (of het menu-item *SGML/Toggle 8-bit insertion*), je kunt ervoor kiezen om de 9-bit tekens in te tikken, of in SGML vorm, b.v. in de vorm **&...;**.

Het laat je met `Meta-x sgml-tags-invisible` (of het menu-item *SGML/Toggle Tag Visibility*) ook de SGML-tags verbergen of tonen.

4.3 PSGML mode

De PSGML mode kan veel bijdragen aan het wijzigen van SGML-documenten met Emacs.

De *psgml-linuxdoc* documentatie geeft een uitleg over hoe je deze mode met *LinuxDoc* kunt installeren en gebruiken.

4.4 Diverse

4.4.1 auto-fill mode

Als je in de normale mode een paragraaf typt en aan het einde van de regel komt, moet je de <Return>-toets gebruiken om naar de volgende regel te gaan, anders komt de gehele paragraaf op één regel. Als je de <Return> gebruikt om naar de volgende regel te gaan, krijg je een paragraaf met ongelijke rechtermarges.

Als je een aantal regels voorbij een redelijke breedte laat gaan, kun je ze met een aantal editors niet zien.

De **auto-fill** mode zorgt dat deze saaie taak automatisch wordt gedaan: als je verder gaat dan een bepaalde kolom (standaard de 70e), ga je automatisch naar de volgende regel.

Je kunt deze mode als volgt gebruiken en de breedte-instelling van je regels op 80 instellen:

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

5 Ispell

Als je vanuit Emacs je document op spelling wilt controleren, kun je daarvoor het **Ispell**-package gebruiken in de Emacs mode.

5.1 Kiezen van je standaard-woordenboeken

Je kunt Emacs zo instelen, dat het bij het laden van een bestand, automatisch woordenboeken selecteert (je kunt er verscheidene gebruiken). De eerste, zeker de belangrijkste, is het hoofdwoordenboek, dat met Ispell wordt gedistribueerd. Je kunt tussen verscheidene talen kiezen. De tweede is je persoonlijke woordenboek, waarin Ispell die woorden zal zetten die het niet in het hoofdwoordenboek kon vinden maar waarvan je aangaf dat ze moesten worden ‘onthouden’.

Als je het Franse woordenboek dat met Ispell wordt meegeleverd, wilt gebruiken als standaard-woordenboek en als je het bestand `.ispell-dico-perso` in je home-directory als een persoonlijk woordenboek wilt gebruiken, zet dan de volgende regels in het bestand `.emacs`:

```
(setq sgml-mode-hook
'(lambda () "Standaardwaarde voor SGML mode."
(setq ispell-personal-dictionary "~/.ispell-dico-perso")
(ispell-change-dictionary "francais")
))
```

5.2 Selecteren van speciale woordenboeken voor bepaalde bestanden

Het kan een probleem zijn als je de spelling in je documenten niet altijd in dezelfde taal wilt controleren. Als je documenten vertaalt, is het zeer waarschijnlijk dat je vaak van taal (en woordenboeken) verwisselt.

Ik weet niet hoe je op enige Lisp-wijze, ofwel automatisch of met een enkele muisklik, het hoofdwoordenboek en persoonlijke woordenboeken geassocieerd met de taal die op dat moment wordt gebruikt, kunt selecteren. (Als je dit wel weet, laat me dit dan alsjeblieft weten!)

Het is echter mogelijk om aan het einde van het bestand aan te geven, welke woordenlijsten je voor het huidige bestand (en slechts deze) wilt gebruiken. Het volstaat om ze als commentaar toe te voegen, zodat Ispell ze bij het opstarten van de spellingscontrole kan inlezen:

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

Als je voorheen, in het bestand `.emacs`, hebt opgegeven dat je standaard-woordenboeken de Franse woordenboeken zijn, dan kun je deze regels aan het einde van ieder bestand in ’t Engels toevoegen.

5.3 Controleren van de spelling in je document

Om je gehele document op spelling te controleren, gebruik je vanuit een willekeurige plaats in het document het commando `Meta-x ispell-buffer`. Je kunt ook slechts de spelling van een deel van je document controleren:

- Markeer het begin van het gebied met `Ctrl-Spc` (mark-set-command),
- Ga naar het einde van het gebied dat moet worden gecontroleerd,
- typ `Meta-x ispell-region`.

Emacs start dan vervolgens Ispell op. Zodra het een onbekend woord tegenkomt, wordt het woord getoond (meestal benadrukt) en wacht tot je één van de volgende toetsen indrukt:

- **spc** accepteer alleen deze keer het woord,
- **i** accepteer het woord en voegt het toe aan je persoonlijke woordenboek,
- **a** accepteer het woord voor deze sessie,
- **A** accepteer het woord voor dit bestand, en voeg het toe aan het lokale woordenboekbestand
- **r** staat je toe om het woord handmatig te wijzigen
- **R** staat je toe om het onjuist gespelde woord in alle voorkomende gevallen te wijzigen,
- **x** stop het controleren en zet de cursor weer terug op z'n plaats,
- **X** stop het controleren en laat de cursor waar het is, zodat je je bestand kunt corrigeren;je zul later weer verder kunnen gaan met het controleren van de spelling als je `Meta-x ispell-continue` intikt,
- **?** geeft je online-help.

Als ispell één of meerdere woorden vindt, die het onbekende woord benaderen, zal het ze in een klein venster tonen, waarbij ieder woord wordt voorafgegaan door een cijfer. Typ gewoon dit cijfer in om het onjuist gespelde woord te vervangen door het corresponderende woord.

5.4 Persoonlijk woordenboek versus lokaal woordenboekbestand

Met de **i** toets kun je een woord invoegen in je persoonlijke woordenboek, en met **A** kun je een woord invoegen in het lokale woordenboekbestand.

Het lokale woordenboekbestand bestaat uit een reeks woorden die aan het einde van het bestand als commentaar worden ingevoegd, en iedere keer dat Ispell met dit bestand wordt opgestart, opnieuw worden ingelezen. Op deze manier, kun je een aantal woorden accepteren, die in dit bestand, maar niet noodzakelijk in andere bestanden zijn toegestaan.

Wat mezelf betreft, denk ik dat het beter is dat het persoonlijke woordenboek gereserveerd moet zijn voor woorden die niet in het hoofdwoordenboek staan maar die wel aan de taal toebehoren (zoals afgebroken woorden), plus een aantal algemene woorden zoals proper nouns of anderen (zoals *Linux*), alz ze niet te veel op een echt woord uit het hoofdwoordenboek lijken; het toevoegen van te veel woorden in het persoonlijke woordenboek zoals voornamen, kan risico's met zich meebrengen omdat ze op een woord uit de werkelijke taal kunnen lijken (je kunt je voorstellen dat Ispell door het volgende wordt gefopt: '*When the going gets tof, the tof get going*[2]

[2] *Tof* is een Franse afkorting voor de voornaam *Christophe*!').

5.5 Controleren van de spelling tijdens het typen

Ispell kan je bestand tijdens het typen op spelling controleren. Hiervoor moet je de **ispell-minor-mode** gebruiken. Om het te starten of te stoppen, tik je in: `Meta-x ispell-minor-mode`. Ispell zal iedere keer dat je een woord intikt en het niet kent een *beep* produceren.

Als die *beeps* je irriteren (of je kamergenoot doet een dutje), kun je die hinderlijke *beeps* vervangen door een flits op het scherm, met het commando `Meta-x set-variable RET visible-bell RET t RET`. Je kunt de volgende regel aan je bestand `.emacs` toevoegen om Emacs voorgoed tot zwijgen brengen:

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

6 Truuks

6.1 Automatisch invoegen van een header

Emacs staat je toe om een aantal acties aan een event te *koppelen* (openen van een bestand, opslaan, opstarten van een nieuwe mode, enz).

De **autoinsert** library maakt van deze faciliteit gebruik: als je een nieuw bestand onder Emacs opent, voegt deze library, overeenkomstig het type bestand, een *standaard* header in.

In ons geval, zou deze *standaard* header evengoed het deel kunnen zijn dat het documenttype (LinuxDoc), de titel, de auteur, en de datum declareert.

Ik zal hier twee manieren beschrijven om een dergelijke header in te voegen. Je zou een sjabloon-bestand met informatie in kunnen voegen, of je zou een **elisp**-routine op kunnen starten.

6.1.1 door het invoegen van een bestand

Je moet Emacs eerst laten weten dat het de `auto-insert` opstart als het een bestand opent, dan de **autoinsert** library in te lezen, die de `auto-insert-alist` lijst declareert die we moeten wijzigen. Deze lijst definieert de header voor ieder bestandstype die moet worden ingevoegd. Standaard moet het bestand dat moet worden ingevoegd in de `~/insert/` directory staan, maar het is mogelijk om de `auto-insert-directory` variabele te herdefiniëren, als je het ergens anders wilt plaatsen.

Voeg de volgende regels toe aan je `.emacs` bestand om het bestand `~/emacs/sgml-insert.sgml` iedere keer dat je een nieuw SGML-bestand opent, in te voegen:

```
(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))
```

Vervolgens kun je in het bestand `~/emacs/sgml-insert.sgml` je aangepaste header wegschrijven en dan Emacs opnieuw opstarten en één of ander bestand `foobar.sgml` openen. Emacs zou je moeten vragen om de automatische invoeging te bevestigen en als je 'yes' antwoordt, je header invoegen.

6.1.2 door het opstarten van een routine

Dit werkt net als voorheen, maar in plaats van de `auto-insert-alist` in te stellen op een bestand dat moet worden ingevoegd, zul je het in moeten stellen op een functie die moet worden uitgevoerd. Zo kunt je verdergaan, ervan uitgaande dat je deze functie wilt wegschrijven in een bestand met de naam `~/emacs/sgml-header.el`. (het is niet nodig je `.emacs` bestand met een dergelijke functie te belasten, aangezien het nogal lang kan worden):

```
(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))
```

In de A (appendix) zul je een voorbeeld aantreffen van de `insert-sgml-header` functie.

A Een insert-sgml-header functie

Deze functie zal voor de gebruiker een aangepaste header, voor een Linux Documentatie Project document, in een nieuw bestand invoegen. Het kan automatisch, wanneer er een nieuw bestand wordt geopend, of expliciet door de gebruiker worden aangeroepen.

Deze functie vraagt de gebruiker, via de *mini-buffer*, om wat informatie, een deel daarvan is nodig, een deel daarvan niet.

Als eerste komt de titel. Als er geen wordt opgegeven, keert de functie onmiddellijk terug, en voegt niets in. Dan komt de datum, de auteur, zijn e-mail en home-page (deze laatste twee zijn optioneel).

Dan komt een verzoek voor de naam van de vertaler. Als er geen is, typ je gewoon *Return*, en er zullen verder geen vragen meer worden gesteld over een hypothetische vertaler. Als er wel één is, wordt je om zijn e-mail en home-pagina gevraagd (ook optioneel).

Deze functie drukt je verzoek vervolgens naar de huidige buffer af, natuurlijk inclusief alle informatie die je in het invoerformulier hebt ingetikt, en ook inclusief de tags die dienen voor de samenvatting en het eerste hoofdstuk. Als laatste zet het de cursor op die plaats waar de samenvatting moet worden ingetypt.

```
(defun insert-sgml-header ()
"Voegt de header voor een LinuxDoc document in"
(interactive)
(let (title author email home translator email-translator home-translator date
              starting-point)
(setq title (read-from-minibuffer "Titel: "))
(if (> (length title) 0)
        (progn
          (setq date (read-from-minibuffer "Datum: ")
                author (read-from-minibuffer "Auteur: ")
                email (read-from-minibuffer "Auteur e-mail: ")
                home (read-from-minibuffer "Auteur home-pagina: http://")
                translator (read-from-minibuffer "Vertaler: "))
          (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 "Vertaler e-mail: ")
            home-translator (read-from-minibuffer "Vertaler home-pagina: http://"))
      (insert "Vertaald door : ")
      (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)
))))
```