

Bli mer effektiv med Emacs!

Andreas Ehliar, Johan Busk Eriksson, Hans Lundmark

26 jan 2010

- 1 Introduktion
- 2 Trevliga finesser
- 3 Makron
- 4 Applikationer
- 5 elisp
- 6 Andra resurser av intresse
- 7 Sammanfattning

Upplägg

- Målgrupp: Vana emacs-användare
- Högt tempo för att hinna demonstrera många finesser
 - Missar du något, kolla filmen eller presentationen på upplysningssidan
 - En enkel .emacs kommer också dit för enkel test av diverse finesser som vi går igenom.

Hjälp

- C-h C-h
- Framförallt: Hjälp om nuvarande mode: C-h m

Rectangular cut & paste

- Markera början av regionen som vanligt (C-SPACE)
- C-x r k Klipp ut
- C-x r y Klistra in
- C-x r t Ersätt med en ny sträng

Krypterade filer

- Om du inte redan har en nyckel, skapa med `gpg --gen-key`
- Spara filen som `fil.gpg`, välj vilken nyckel du vill använda.
- Lägg lämpligtvis in en `-*- foobar -*-` i din fil

generic-mode

- Problemställning: Ditt favoritfilformat stöds inte av emacs
- Lösning: generic-mode

```
(define-generic-mode ssh-config
  '( "#" ) ; Kommentarsinledning
  '( "Host" ; Lista med nyckelord
    "Port"
    "HostName"
    "IdentityFile" )
  nil ; Regexp-nyckelord
  '( "\\\\.ssh/config" ) ; Filer vi matchar till
  nil
  "Generic mode for SSH config files")
```

Generic mode: Regexp-fältet

```
(define-generic-mode ssh-config
  '( "#" )
  '( "Host" "Port" "HostName" "IdentityFile" )
  nil
  '( "\\ssh/config" )
  '( ( "[0-9]+\\. [0-9]+\\. [0-9]+\\. [0-9]+ " .
      'font-lock-warning-face)
    "Generic mode for SSH config files" )
```


re-builder

- Bygg interaktiva regexp M-x re-builder
- Hitta nästa träff: C-c C-s
- Hitta föregående träff: C-c C-r
- Avsluta: C-c C-q

Ediff

- M-x ediff-files eller M-x ediff-buffers
- Hjälp: ?
- Byt fönsterlayout: |
- Scrolla ner/upp: v/V
- Gå till nästa skillnad: n
- Gå till förra skillnaden: p
- Avsluta: q

Emerge

- M-x ediff-merge-files
- Fungerar som ediff men med fler finesser
- Välj ändring från buffer a: a
- Välj ändring från buffer b: b
- Spara buffer c: w c

cua-mode

- För de som är vana/vill använda "windowkortkommandon"
 - C-x, C-c, C-v – klipp ut, kopiera, klistra in
(Aktivt vid markeringar, annars konflikt med andra kortkommandon)
 - Shift-piltangenter för att markera områden

- För att aktivera:

```
(cua-mode t)  
(transient-mark-mode 1) ; Se markerat omr.
```

- Mer info:
 - info Emacs CUA Bindings
 - <http://www.emacswiki.org/CuaMode>

kill-ring 101

- Kill-ring innehåller de 60 senast "dödade" elementen.
- C-w dödar region, C-k till slutet av raden du är på, M-k dödar en mening
- Det som "dödas" hamnar först i kill-ring
- C-y yankar ut det första elementet i kill-ring
- Vill man yanka något annat i kill-ring kan man därefter cycla med M-y

Align:a text

- M-x align-regexp <RET> REGEXP <RET>
 - Exempel

fill-paragraph, auto-fill-mode

- fill-paragraph
fill-paragraph (M-q) radbryter paragrafen till 70 tecken (värdet på fill-column)
 - Med prefix-argument (t.ex. C-u M-q) så marginaljusteras paragrafen också.
- auto-fill-mode
 - Om man vill radbryta all text i en buffer när man editerar den.
 - Som hook till L^AT_EX-mode (auctex), t.ex:

```
(add-hook 'LaTeX-mode-hook 'turn-on-auto-fill)
```

- auto-fill på kommentarer i c-mode

```
(add-hook 'c-mode-hook (lambda ()
  (auto-fill-mode 1)
  (setq comment-auto-fill-only-comments t)))
```

Citera tecken

- Tabb ofta bundet till någon funktion
 - C-q tab
- Även för tecken som saknas på tangentbordet
 - C-q 007 -> "bell"

iswitchb, ido-mode

Alternativ till C-x b (switch-to-buffer)

- Bufferurval minskas vartefter man skriver

```
(iswitchb-mode 1)
```

- ido-mode (interactively do things) är en utvecklad version av iswitchb
 - Ändrar även beteendet för t.ex. find-file (C-x C-f) till att vara mer interaktiv.

```
(require 'ido)  
(ido-mode t)
```

emacs-server+daemon

- Samma buffers i tty och X
- Snabbstartat
- Starta emacs-daemon
 - `$ emacs --daemon`
 - `export ALTERNATE_EDITOR=""`
- emacsclient
 - `-c` skapar ny frame (X)
 - `-n` väntar inte på att du ska "vara klar" med emacs (bra om man startar med `-c` i terminal)
 - `-t` startar en terminalemacs i den terminal du skriver in det
- Skapa kortare alias eller script för att starta klienter
- Tips: skriv init-filer som startar/stoppar emacs `--daemon` ;)

Stavningskontroll

- Flyspell stavningskontrollerar ord allteftersom de skrivs
 - Använder aspell/ispell
 - flyspell-prog-mode kan stavningskontrollera kommentarer i kod
- ispell
 - Har inbyggt stöd i emacs
 - M-x ispell
 - M-\$

Tangentbordsbindningar

(C-h k [tangent] för att ta reda på vad tangenten heter)

- Globala tangentbordsbindningar

Exempel

```
(global-set-key (kbd "<f9>") 'funktion)
```

- Mode-lokala
Lämpligast att definiera i hooks
 - Generellt
(local-set-key (kbd "C-c C-t") 'funktion)
 - CC-modederiverade modes
(define-key c-mode-base-map [f10] 'step-over)

Hooks

Sätt att hooka in funktioner när t.ex. ett mode laddas.

- T.ex.

```
(add-hook 'xxx-mode-hook 'funktion)
```

(Funktionen kan också vara en lambda-funktion)

Windmove

- Flytta markören mellan olika fönster enklare
 - Shift+piltangenter istället för C-x o
- Aktivera:

```
(windmove-default-keybindings)
```

M-piltangenter istället för shift:

```
(windmove-default-keybindings 'meta)
```

msf-abbrev

<http://www.bloomington.in.us/~brutt/msf-abbrev.html> Installation:

- Lägg msf-abbrev i load-path

```
(require 'msf-abbrev)
(setq msf-abbrev-verbose t) ;; optional
(setq-default abbrev-mode t)
(setq save-abbrevs nil)
(setq msf-abbrev-load-tree "~/emacs.d/mode-abbrevs")
(setq msf-abbrev-root "~/.emacs.d/mode-abbrevs")
(global-set-key (kbd "C-c q")
  'msf-abbrev-goto-root)
(global-set-key (kbd "C-c w")
  'msf-abbrev-define-new-abbrev-this-mode)
(msf-abbrev-load)
```

msf-abbrev

- C-c w låter dig nu skapa nya abbreviations.

```
for (<query "iterator? "> = 0;  
     i < <query "max? ">; i++) {  
<cursor>  
}
```

- När jag sen skriver det jag namngav förkortningen till kommer detta expanderas till filens innehåll
- C-c q öppnar katalogen med abbrevs i en dired-buffer

Tramp

Transparent sätt att redigera filer remote C-x C-f
/user@host:filenamn

- Även lokala filer som root:
 - C-x C-f /sudo::/etc/hosts
 - C-x C-f /su::/etc/hosts
 - :: -> root, för andra användare, skriv user@localhost mellan ::

whitespace-mode

- Visualiserar whitespace-tecken (blanksteg, radbrytningar, tabb)
- Kan visa rader som går över 80 tecken (whitespace-line-column) med annat face
- (M-x delete-trailing-whitespace)

Inkrementell sökning

- De flesta här känner nog till C-s
- Det finns en hel del mindre kända options till isearch, exempelvis:
 - Togglar regexpsökning: M-r
 - Lägg till ordet under markören till söksträngen: C-w
 - Editera sökning i minibuffern: M-e
 - Highlighta söksträngen: M-s h r
 - Lista alla sökträffar i separat fönster: M-s o
 - Hämta tidigare sökning: M-p

Shell-kommandon i emacs

Shell-kommandon används av emacs i bland annat:

- M-x grep
- M-| (M-x shell-command-on-region)
- M-x compile (mer om detta senare)

M-x shell

Ditt favoritskal inuti emacs.

- Förra kommandot: M-y
- Nästa kommando: M-p
- Kräver enligt mig lite konfiguration för att bli användbart:

```
(custom-set-variables
 '(comint-scroll-to-bottom-on-input t)
 '(comint-scroll-to-bottom-on-output t)
 '(comint-scroll-show-maximum-output t)
 '(comint-completion-autolist t)
 '(comint-completion-addsuffix t)
 ... ; other settings for Custom
)
```

M-x compile

- Visar kompileringsloggen i en emacs-buffer
- Öppna enkelt den fil på den rad som ger fel- eller varnings-meddelande
- Upprepa kompileringen M-x recompile

M-x highlight-changes-mode

- Visar vad du har ändrat i filen
- Visa/dölj ändringar: M-x highlight-changes-visible-mode

Introduktion

- EMACS - Editing MACroS
- Börja makro: F3 eller C-x (
- Avsluta makro: F4 eller C-x)
- Kör makro: F4 eller C-x e
- Kör makro och fortsätta editera: C-u F3

Standardtips för effektivare makroanvändning

- Om möjligt, börja makrot med exempelvis C-a eller C-e
- Flytta markören via exempelvis C-a, C-e, M-f, M-b eller C-s
- Undvik att flytta markören via piltangenterna
- Undo går bra att använda i makron

Namnge, spara och editera makron

- M-x name-last-kbd-macro
- M-x insert-kbd-macro
- M-x edit-kbd-macro

Keyboard macro counter

- C-x C-k C-c Sätt räknaren till ett visst värde
- F3 Lägg in räknaren

Diverse andra tips

- Det kan vara kraftfullt att kombinera makron med användning av flera buffrar
- Kör ditt makro tills något går fel: C-u 0 M-x makronamn

Emacs istället för fula shell-script med sed, cut, awk, etc

- Emacs är inte så dumt för exempelvis textfiltrering/formattering
- Skapa lämpligt makro som editerar filen som du vill ha den
- emacs fil.txt -f mittfinamakro -f save-buffer -f kill-emacs

Dired

- Öppna ett bibliotek med C-x C-f
- Killer feature: M-x wdired-change-to-wdired-mode
- Editera buffern!
- C-c C-c för att byta namn på allt
- Det finns även många andra finesser i dired. . .

org-mode

- I grund och botten en utökning av outline-mode
- Bra för strukturerade anteckningar
- Ett stort antal avancerade finesser

org-mode

- Min favoritfeature: Schemaläggning
- Schemalägg händelse: C-c C-s
- Schemalägg sista datum: C-c C-d

org-mode

- Visa veckans agenda: C-c a a
 - (Beror lite på vad du bundit org-agenda till)
- Markera något som gjort i agendan: t
- Markera något som gjort i en org-fil: C-c C-t

org-mode och remember

- Det är ofta drygt att öppna sin .org-fil bara för att göra en enkel anteckning.
- M-x remember (binda detta till något bra kortkommando)
- Skriv in din uppgift, schemalägg den om nödvändigt, sedan C-c C-c
- Remember lägger automatiskt in en länk till aktuellt dokument
 - Integrerat med epost-läsare i emacs

org-mode och remember

- Det går även att göra olika mallar för remember
 - Att göra
 - Möten
 - Plötsliga bra idéer
 - (Plötsliga mindre bra idéer)

org-mode

- Kraftfullt verktyg för att skapa utkast och mindmaps, men även "riktiga" dokument.
 - (Denna presentation är gjord i org-mode och exporterad till \LaTeX /PDF)
- Tabeller
 - Enkla kalkylark
 - Exporterbara

Calc

- Avancerad stackbaserad miniräknare i emacs. Kan rita grafer med hjälp av gnuplot.
- Starta med 'M-x calc'.
- Skriven av Dave Gillespie 1990–93 (helt i elisp).
- Den gamla versionen (2.02f) distribueras fortfarande med XEmacs. (Några små uppdateringar av kompatibilitetsskäl, de senaste 2004.)
- Calc underhålls sedan 2001 inom ramen för GNU-projektet.
- En uppdaterad version medföljer GNU Emacs version 22.3 och högre. (Inga stora ändringar här heller, men man har t.ex. infogat en "Calc"-meny i menyraden, vilket är praktiskt eftersom det kan vara svårt att komma ihåg de många tangentbindningarna.)

Datatyper

- Godtyckligt stora heltal.
- Rationella tal (dvs bråk).
- Flyttal (i basen 10 internt) med godtycklig precision.
- Symboliska uttryck!
- Komplexa tal, matriser, intervall, datum, enheter, m.m.

Att använda calc i calc-bufferten

i	(info)	Öppna manualen i info-format.
U	(undo)	Ångra.
q	(quit)	Avsluta (gömmar calc-fönstret, men bufferten finns kvar).

1 2 3	SPC/RET	Talet 123 läggs på stacken.
SPC/RET	ensamt	Översta talet på stacken dupliceras.
TAB		De två översta talen på stacken byter plats.
M-TAB		Tredje talet från toppen flyttas överst.
DEL		Översta talet på stacken tas bort.

Många andra kommandon finns för att komma åt eller flytta data på stacken.

Att använda calc i calc-bufferten

- + De två översta talen på stacken tas bort, och deras summa läggs på stacken istället.
(Analogt för övriga räknesätt: $-$ $*$ $/$ $^$ $.$)
- n Negation: talet x överst på stacken ersätts med $-x$.
- Q Kvadratrot (sQuare root): x ersätts med \sqrt{x} .
- E/L/S/C x ersätts med e^x / $\ln x$ / $\sin x$ / $\cos x$.
- I S x ersätts med $\arcsin x$ (prefix 'I' som i *Invers*).

Symboliska uttryck

Apostrof \Rightarrow Inmatning av formel i minibufferten:

' x SPC y ^ 2 - 4 RET Formeln $xy^2 - 4$ läggs på stacken.

Prefix 'a' = ALGEBRAIC.

- a s (simplify) Förenkla.
- a d Derivera (m.a.p. den variabel som calc frågar efter i minibufferten).
- a x (eXpand) Utveckla (multiplicera ut parenteser etc.).
- a f Faktoriserar polynom. (För heltal, använd 'k f'.)

Inställningar

- p Ändra flyttalsprecision (calc frågar efter nytt värde i minibufferten).

Prefix 'm' = MODE. Massor av inställningar, i stil med:

m f	(fractions)	Växla mellan $1/2 = 1:2$ (bråk) och $1/2 = 0.5$ (flyttal).
m d	(degrees)	Mät vinklar i grader.
m r		Mät vinklar i radianer.
m m		Spara inställningar i filen <code>~/.calc.el</code> för framtida sessioner.

Mera inställningar

Prefix 'd' = DISPLAY.

d N	(normal)	Standardläge.
d B	(big)	"Prettyprinting".
d O	(one-line)	Matriser visas på en enda rad; lättare att klippa och klistra.
d b	(break)	Slå av/på radbrytning i stacken.
d l	(line numbers)	Slå av/på radnumrering i stacken.
d C/F/L/W/M/...		Använd formelsyntax från C / Fortran / \LaTeX / Maple / Mathematica / ...

Matriser

- Radvektor = lista = [1,2,3]
- Matris = lista av radvektorer = [[1,2,3],[4,5,6],[7,8,9]]

Prefix 'v' eller 'V' = VECTOR. T.ex.:

5 v p	(vector pack)	Packa de 5 översta elementen på stacken i en vektor.
v u	(vector unpack)	Tvärtom.
v t		Transponera matris.
V M <key>	(vector map)	På vektorn överst på stacken utförs elementvis den operation som är bunden till <key>. Man kan välja <key>=apostrof och mata in en funktion (med \$ på den oberoende variabelns plats, t.ex. "\$ ² +3").

Variabler

Prefix 's' = STORE

- s s (store) Lagra uttrycket överst på stacken i en variabel (calc frågar efter variabelnamnet i minibufferten).
- s t Lagra, och ta sedan bort värdet från stacken.
- s r (recall) Hämta värdet av en variabel och lägg överst på stacken.
- s u (unstore) Återställ variabel till ursprungligt ("tomt") tillstånd.

- s + Addera värde till variabel. (Analogt för övriga räknesätt.)
- = Varje variabel i uttrycket överst på stacken byts ut mot sitt lagrade värde ("tomma" variabler lämnas orörda).

Variabler

- s e (edit) Redigera en variabels värde i en temporär buffert.
- s l (let) Calc frågar efter formel (av typen "x=5") och substituerar det angivna värdet i uttrycket överst på stacken. Värdet lagras ej permanent i den angivna variabeln.

- Fördefinierade variabler (med "magiska" värden som evalueras med den valda flyttalsprecisionen): *pi*, *e*, *phi* (gyllene snittet), *gamma* (Eulers konstant).
- Även *i* (imaginära enheten, $i^2 = -1$) är lite magisk.

Selection mode

En finess i calc som saknas i många andra datoralgebrasystem är möjligheten att gå omkring med markören i formler och operera på markerade deluttryck (prefix 'j').

j s	(select)	Markera minsta möjliga deluttryck runt markören.
j m/l	(more/less)	Utvidga/krymp markeringen.
j u	(unselect)	Avmarkera den formel som markören står på.
j c	(clear)	Avmarkera allt på hela stacken.
j d	(display)	Växla mellan att visa det markerade eller allt som <i>inte</i> är markerat.

Selection mode

j n/p	(next/previous)	Flytta markeringen.
j R/L	(right/left)	"Dra" det markerade uttrycket åt höger/vänster i formeln.
j C	(commute)	Byt plats på termer.
j D	(distribute)	Multiplicera in i parenteser, m.m.
j M	(merge)	Motsatsen till 'j D'.

Reglerna som styr 'j D' och 'j M' finns i variablerna `DistribRules` respektive `MergeRules`, som man kan modifiera med egna regler om det skulle behövas. Exempel på hur distributionsregler kan se ut:

- $x * \text{select}(a + b) := x * \text{select}(a) + x * b$,
- $x / \text{select}(a + b) := 1 / (\text{select}(a)/x + b/x)$
- $\exp(\text{select}(a + b)) := \exp(\text{select}(a)) * \exp(b)$

Att använda calc från andra buffertar

- Det finns flera kommandon för att flytta data mellan calc-bufferten och andra buffertar.
- Man kan även använda calc för att manipulera uttryck direkt i t.ex. en C-fil eller en \LaTeX -fil. Prefix för sådan calc-relaterade kommandon: 'C-x *' (förut 'M-#').
- Aktivera och avaktivera med 'C-x * e' (embed).
- Calc gissar lämpligt sätt att identifiera formler baserat på buffertens mod (t.ex. $\$. \dots \$$ för \TeX -filer).
- Den formel som markören står på läggs överst på stacken (som inte syns, såvida inte calc-fönstret är öppet; 'C-x * o' öppnar det), och uppdateras "live", dvs alla tangenttryckningar tolkas som calc-kommandon så länge "embedded mode" är aktivt.
- Tabeller i org-mode + calc = kalkylark!

Wanderlust

- Epostklient, främst för IMAP, men även POP, NNTP och MH/Maildir
<http://www.gohome.org/wl/>
- Kan kopplas till orgmode (remember) och bddb (adresser)

WL konfigurering

| .emacs:

```
(autoload 'wl "wl" "Wanderlust" t)
(autoload 'wl-other-frame "wl" "WL on new frame." t)
(autoload 'wl-draft "wl-draft" "Write draft with WL." t)

;; Set wandelust as mail-user-agent
(autoload 'wl-user-agent-compose "wl-draft" nil t)
(if (boundp 'mail-user-agent)
    (setq mail-user-agent 'wl-user-agent))
(if (fboundp 'define-mail-user-agent)
    (define-mail-user-agent
      'wl-user-agent
      'wl-user-agent-compose
      'wl-draft-send
      'wl-draft-kill
      'mail-send-hook))
```

WL konfigurering (~/.wl)

```
;; -*- mode: emacs-lisp -*-  
  
;; Set main IMAP server  
(setq elmo-imap4-default-server "imap.lysator.liu.se")  
(setq elmo-imap4-default-port 143)  
(setq elmo-imap4-default-stream-type 'starttls)  
  
;; Setup SMTP  
(setq wl-smtp-connection-type 'starttls)  
(setq wl-smtp-posting-port 587)  
(setq wl-smtp-authenticate-type "plain")  
(setq wl-smtp-posting-user "user@lysator.liu.se")  
(setq wl-smtp-posting-server "mail.lysator.liu.se")  
(setq wl-local-domain "lysator.liu.se")
```

WL-konfigurering (~/.wl) II

```
;; Ha BCC i draft  
(setq wl-bcc "")
```

```
;; Kontrollera om nya mail ankommit valda folders varje  
(setq wl-biff-check-folder-list '("%INBOX" "%foo-list"))  
(setq wl-biff-check-interval 900) ;; sekunder  
(setq wl-biff-notify-hook '(ding)) ;; beep on mail
```

WL-konfigurering (~/.folders)

```
%INBOX  "Inbox "  
%Sent   "Sent "  
%Trash  "Trash "  
Listor {  
    %upplysning      "UppLYSning "  
    %eff              "EFF "  
}
```

- Använd hellre M-x wl-fldmgr-* för att hantera av folders

Lösenord

Om man vill slippa skriva in lösenord varje gång man vill skicka/ta emot post

- M-x elmo-passwd-alist-save
- M-x elmo-passwd-alist-clear (för att ta bort sparade lösenord, spara också alist efteråt)

Gnus

- Kombinerad mail och newsläsare
- Klarar att läsa mail från exempelvis mbox, maildir och IMAP
- GNUS är tyvärr lite segt när man har för mycket mail
 - Självt använder jag gnus + maildir + offlineimap
 - Andra använder gnus + imap + dovecot

Snabb intro till GNUS

- Inte svårt att testa GNUS på en Maildir-folder

```
(setq gnus-select-method '(nnil ""))  
(setq gnus-secondary-select-methods  
      '((nnmaildir ""  
          (directory "~/upplysning/demofiler/mail/")))))
```

GNUS - snabb intro

- Markera mail för borttagning: E
- Flytta mail till annan mapp: B m
- Svara på mail: r
- Egentligen förtjänar GNUS en egen upplysning. . .

Några snabba elisp-tips

- En hel del av dessa applikationer behöver konfigureras med elisp
- Mycket elisp-utveckling kan göras utan att emacs behöver startas om
- Bra knappkombinationer:
 - Evaluera föregående elisp-uttryck: C-x C-e
 - I lisp-interaction-mode kommer C-j att evaluera och skriva ut resultatet av lisp-uttrycket i buffern

Mer inspiration kan hittas på exempelvis:

- <http://www.emacswiki.org/>
- <http://emacs-fu.blogspot.com/>
- <http://emacs.wordpress.com/>
- <http://orgmode.org/worg> ("wiki" genererad från org-mode-filer, om org-mode)
- <http://emacsblog.org/>
- Lyskom-mötet om emacs
 - Posta era favorittips där efter upplysningen :)

Sammanfattning

```
(animate-string (concat  
  "Vi hoppas att ni blivit inspirerade "  
  "att leta rätt på lite nya finesser "  
  "i vår favoriteditor!") 5 5)
```