

# Introduktion till emacs

---

Introduktion till emacs  
Av:  
Kjell Enblom

Denna presentation är fri och lyder under GNU Free Documentation

License version 1.1 eller senare.



# Presentationen

---

- Hela denna presentation finns på:  
<http://www.lysator.liu.se/~kjell-e/tekla/linux/dokument.html>
- David Kågedals emacs-lathund finns på:  
<http://www.lysator.liu.se/upplysning/emacs/lathund.ps>



# Vad är emacs?

---

- Emacs är en kraftfull editor.
- Emacs kan bland annat användas till
  - ◆ för att editera texter
  - ◆ skriva rapporter (min exjobbssrapport är gjord med emacs och LaTeX)
  - ◆ göra presentationer (den här)
  - ◆ för programutveckling
  - ◆ till att läsa e-post
  - ◆ till att läsa news
  - ◆ ...



# Hur det började

---

- Emacs utvecklades från början på 1970-talet i redaktorn TECO, Text Editor and Corrector
- Richard M. Stallman var på besök på Stanford och såg editorn E.
- Inspirerad av E utvecklade Stallman editorn som han gav namnet emacs.
- Namnet stod för "Editing MACroS" eller "E with MACroS".
- TECO var inte fri. Bland det första Stallman gjorde efter att ha grundat FSF var att utveckla en fri emacs.



# Emacs

---

- Förutom den vanliga emacs finns även xemacs.
- Emacs finns förutom till unix även till andra system som bland annat Windows.
- Emacs kan köras i ett terminalfönster eller eget X-fönster.



# Utbyggbar

---

- Emacs är utbyggbar i lisp-dialekten elisp.
- Mycket i emacs är gjort i elisp.
  - ◆ Webbläsare
  - ◆ Epost-program (RMAIL, miu ...)
  - ◆ News-program (gnus)
  - ◆ Planerare (planner)
  - ◆ Miljöer för de flesta programspråk
  - ◆ Filhanterare
  - ◆ Ftp-klient
  - ◆ LysKOM-klient



# Dokumentkonventioner

---

- I emacs-dokumentation skrivs control-tangenten (<ctrl>) följt av en tangent som C-x (där ctrl i det här fallet följs av x).
- Meta följt av ett tecken skrivs som M-x (där meta i det här fallet följs av x).



# Starta emacs

---

- Emacs kan startas från ett terminalfönster med kommandot:

```
emacs
```

- Ofta startar man emacs i bakgrunden på ett fönstersystem:

```
emacs &
```

- Ibland kan man dock vilja starta emacs i terminalfönstret istället

```
emacs -nw
```





# Några flaggor

---

- När man startar emacs går det att ge många flaggor.
- -l fil.el laddar in och kör elispfilen fil.el
- -f funktion startar emacsfunktionen, t.ex. -f lyskom för att starta LysKOM-klienten.
- -fn FONT startar emacs med typsnittet FONT.
- -title foobar sätter namnet i fönstrets titelrad till foobar.
- -q gör så att emacs inte läser in filerna .emacs och default.el.



# Kortkommandon

---

- För emacs används control-tangenten (<ctrl>) och alt-tangenten eller escape (<meta>) för i stort sett alla kortkommandon.
- För <ctrl> trycker man ner <ctrl>-tangenten och nästa tangent samtidigt.
- För <meta> så trycker man ner alt-tangenten och nästa tangent samtidigt.
- Ett alternativ för <meta> är escape-tangenten som man trycker ner och släpper före nästa tangent.



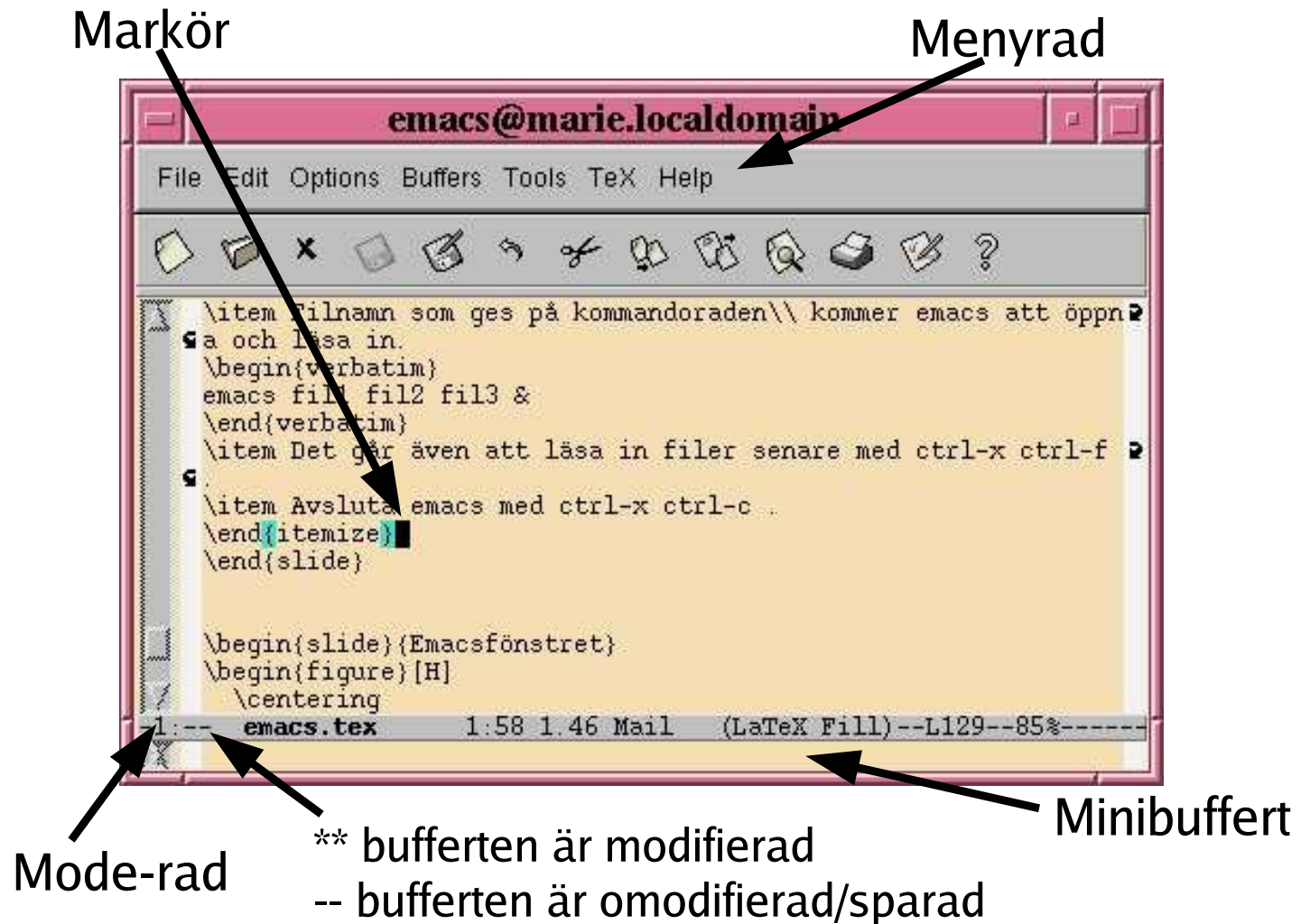
# Starta och avsluta emacs

- Först läser emacs in och kör filen `~/.emacs`
- Filnamn som ges på kommandoraden kommer emacs att öppna och läsa in.  

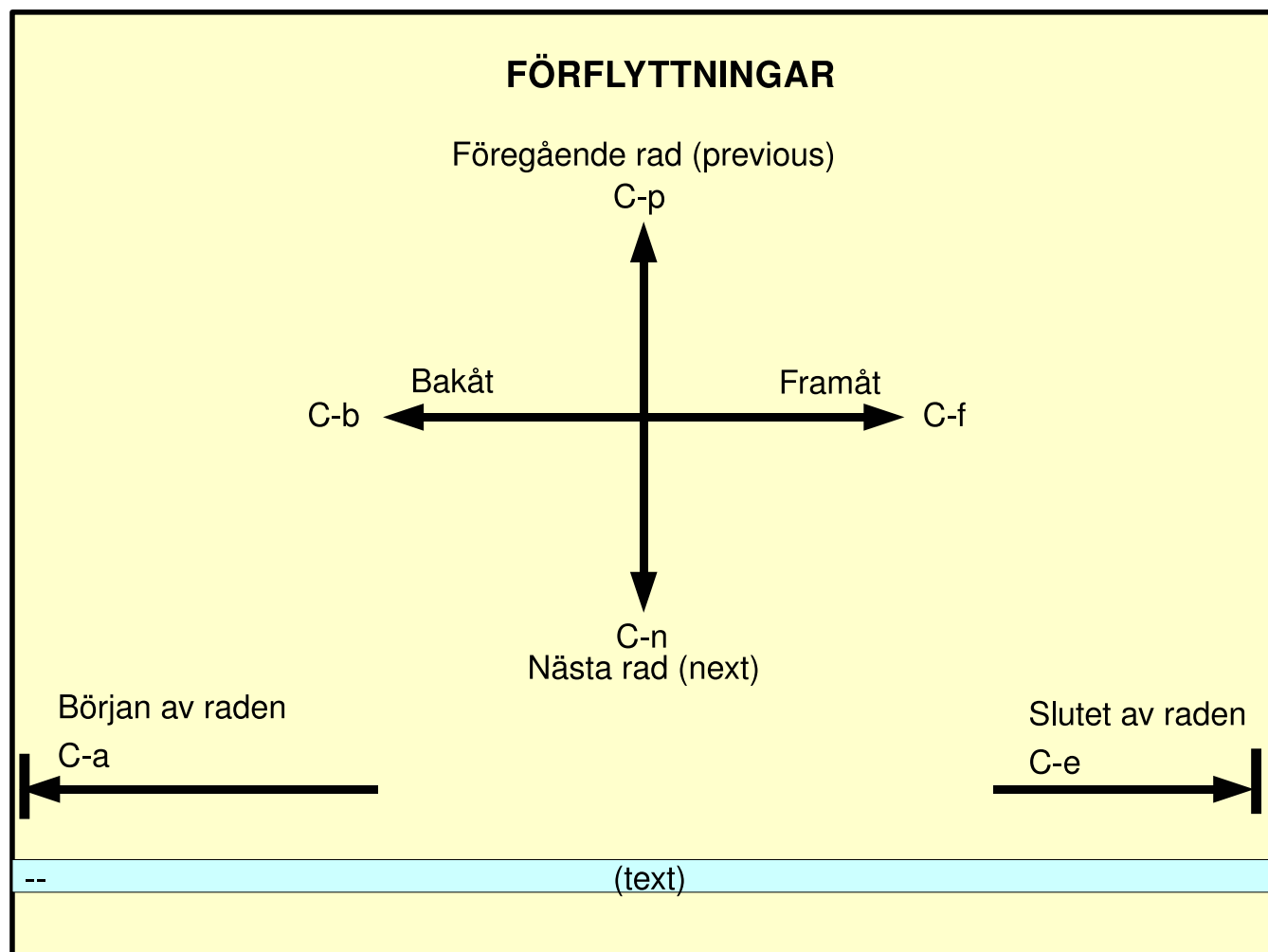
```
emacs fil1 fil2 fil3 &
```
- Det går även att läsa in filer senare med `C-x C-f`.
- Avsluta emacs med `C-x C-c`.



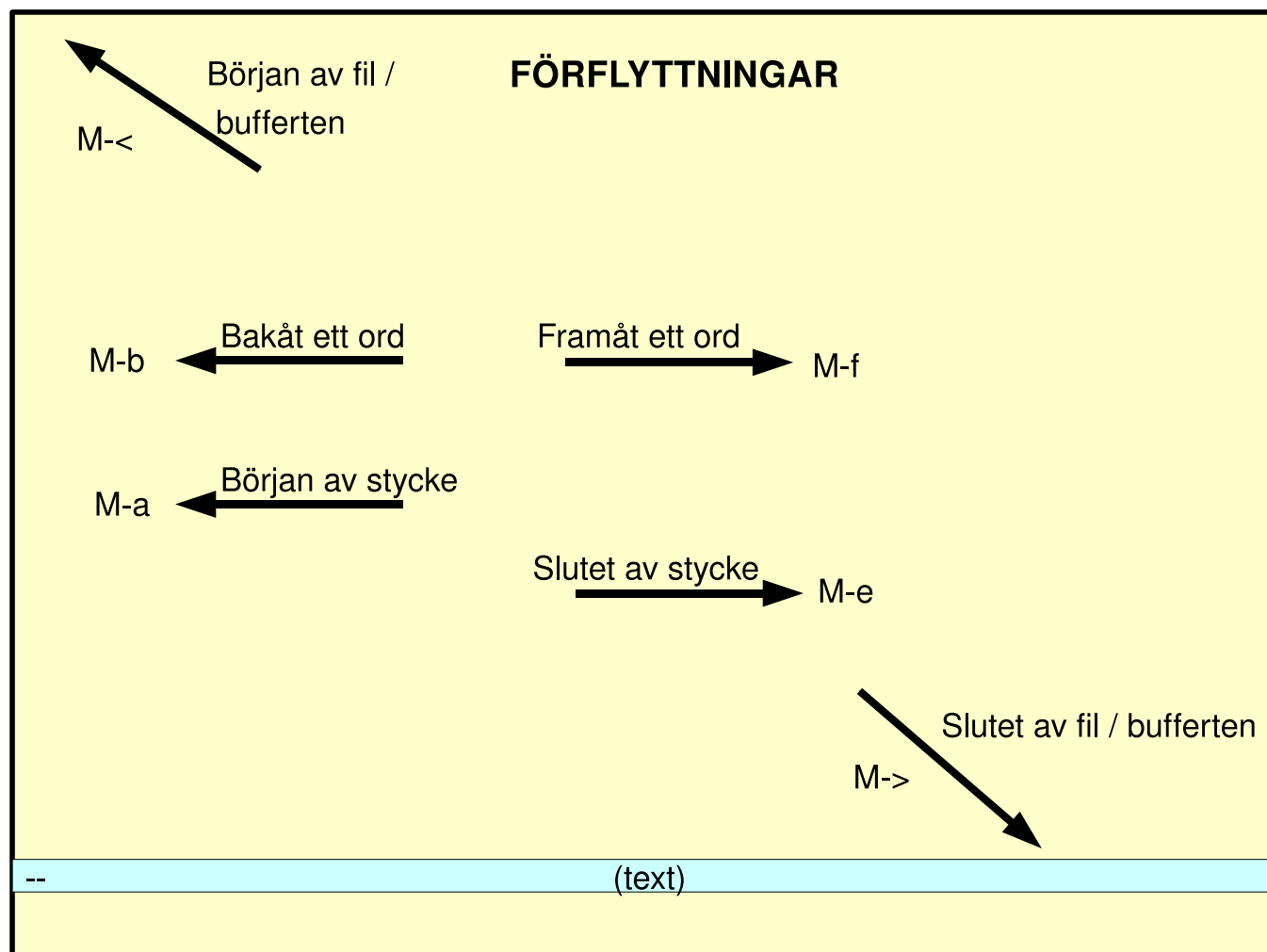
# Emacsfönstret



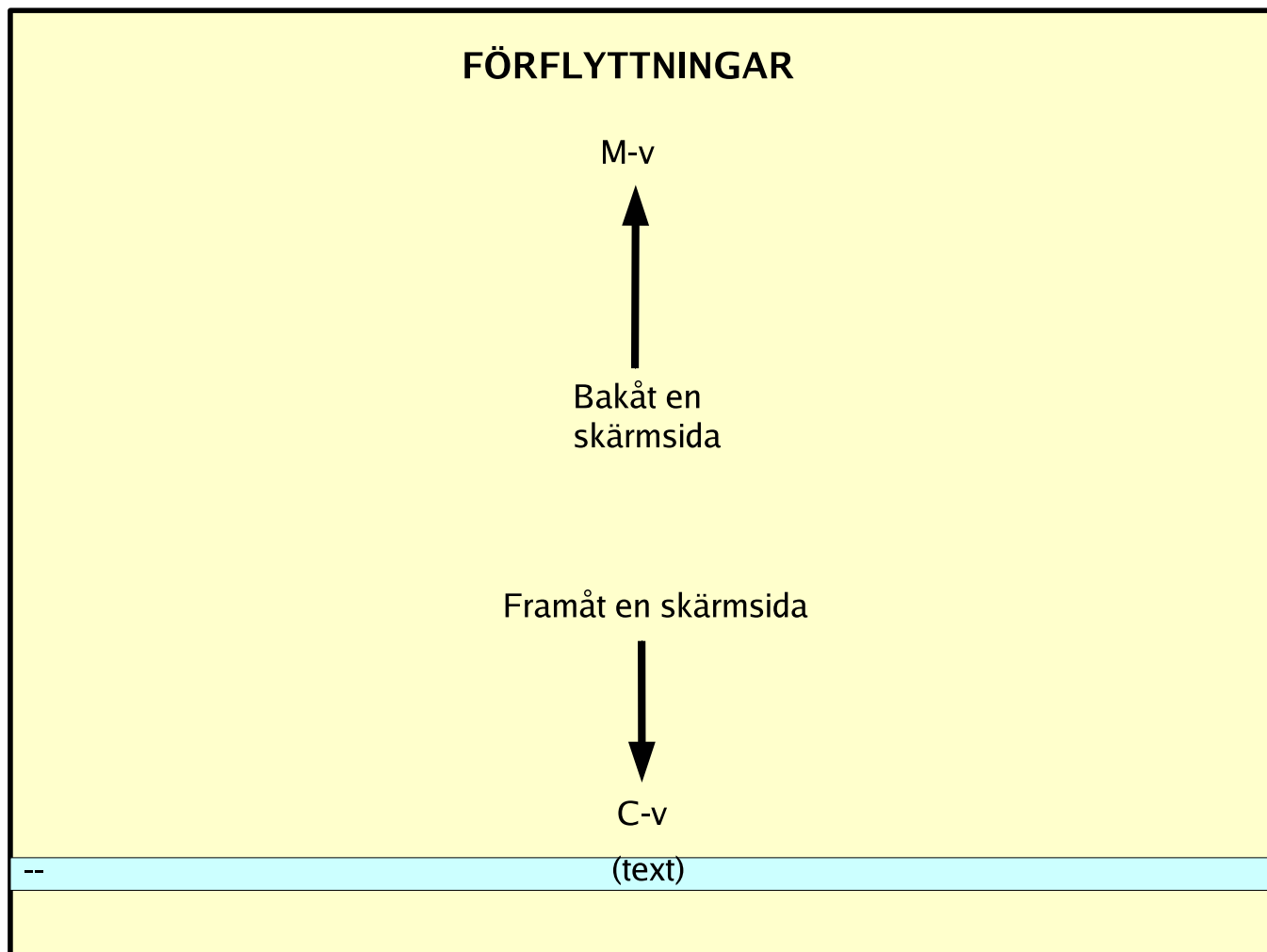
# Förflyttningar



# Förflyttningar



# Förflyttningar



# Förflyttningar

---

- Det går även att använda piltangenterna och PageUp och PageDown.
- Med C-l (bokstaven L) ritar emacs om fönstret och centrerar kring raden där markören står.





# Avbryta kommandon

---

- Om man har påbörjat ett kommando och vill avbryta det så görs det med C-g (eventuellt två C-g efter varandra).
- Om man befinner sig i minibufferten längst ner i emacs går det att komma ur med `<esc><esc><esc>`.



# Redigering

---

- Beroende på hur tangentbordet är mappat så raderar <backspace> ett tecken bakåt och <delete> resp C-d tecknet som markören står på.
- C-k klipper ut från markören till slutet av raden.



# Redigering

---

- M-<backspace> raderar ordet till vänster om markören.
- M-d raderar ordet till höger om markören.



# Redigering

---

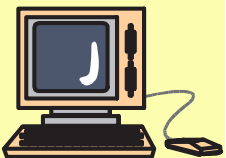
- En text markeras med:  
Markera en punkt med C-<mellanslag> i början eller slutet. Gå till andra änden.
- Den markerade texen benämns som region.
- En markerad text klipps ut med C-w.
- En markerad text kopieras med M-w.
- En urklippt/kopierad text klistras in med C-y (yank).
- Vanligtvis följer punkten (point) markören men C-<mellanslag> sätter en punkt på annan plats.



# Redigering

---

- C-t byter plats på två tecken, tecknet markören står på och tecknet före.
- M-t byter plats på två ord, ordet markören står på och ordet efter.



# Ångra

---

- Ångra går att göra med M-x undo C-x u eller med C-\_ .
- Om undo upprepas så ångras den näst senaste ändringen o.s.v.
- Om man gör undo för mycket går det att flytta på markören och trycka undo så görs det en redo, d.v.s. en undo på sista undo.
- Upprepning av redo gör undo på tidigare utförda undo.



# Urklippsringen

---

- Urklippen sparas i en urklippsring
- <ctrl->-y klistrar in det senaste utklippta.
- Med M-y efter C-y klistrar man in det näst senaste utklippta. Observer att markören inte får flyttas.
- M-y kan upprepas för att få tidigare urklipp.
- M-y kan upprepas tills dess att man är tillbaka till det senaste.



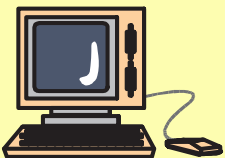
# Sök

---

- Söka framåt görs med C-s. I minibufferten står då

I-search:

- Emacs börjar söka så fort du skriver in tecken.
- För att söka efter nästa förekomst tryck på C-s igen.
- För att söka bakåt tryck C-r (reverse) och gör i övrigt som för framåtsökning.





# Sök

---

- En sökning kan avbrytas med C-g som gör att man kommer tillbaka till början av sökningen.
- En sökning kan avbrytas genom att t.ex. gå ett steg framåt eller bakåt.



# Sök och ersätt

---

- Sök och ersätt kan göras med M-%
- Skriv in ord som ska ersättas.
- Skriv in ord som det ska ersättas med.
- Tryck på y eller <mellanslag> för att ersätta.
- Tryck på n för att inte ersätta.
- Tryck på ! för att ersätta alla förekomster.
- Sök och ersätt sker från markören och framåt.



# Spara respektive läsa in en fil

---

- En fil kan läsas in med C-x C-f följt av sökväg och filnamn i minibufferten.
- En fil sparas med C-x C-s.
- Alla osparade filer / buffertar sparas med C-x s.
- Spara som görs med C-x C-w följt av sökväg och filnamn i minibufferten.



# Buffertar

---

- När en fil läses in i emacs hamnar den i en buffert.
- Varje fil har en egen buffert.
- Förutom buffertarna för filerna finns ytterligare några buffertar. Dessa är bland annat *\*scratch\**, *\*Completions\** och *\*Messages\**
- Varje buffert har ett unikt namn.
- Buffertens namn visas i mode-raden.



# Buffertar



Buffertens  
namn

Mode-rad

\*\* bufferten är modifierad

-- bufferten är omodifierad/sparad

Minibuffert



# Buffertar

---

- Man byter buffert med C-x b följt av buffertens namn.
  - ◆ Det går att skriva ? för att få upp en lista över alla buffertar.
- Buffertar kan listas med C-x C-b
- Aktuell buffert kan dödas med C-x k



# Fönster

---

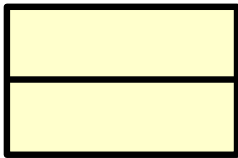
- Emacs kan delas upp i flera fönster.
- Det går att visa samma eller olika buffertar i de olika fönstren.
- Varje buffert har en egen mode-rad.
- Genom att visa samma buffert i olika fönster kan man samtidigt studera olika delar av en buffert.



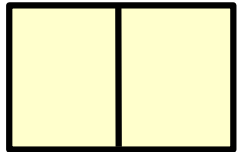
# Fönster

---

- C-x 2 delar ett fönster i två horisontellt



- C-x 3 delar ett fönster i två vertikalt



- För att hoppa till nästa fönster trycker man C-x o (other window).
- För att döda ett fönster ställer man sig i det och trycker C-x 0 (OBS! en nolla)

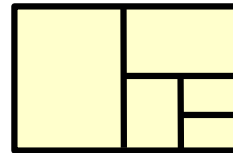




# Fönster

---

- För att återgå till ett enda fönster trycker man C-x 1
- Med C-x 2 och C-x 3 går det att få många fönster i varandra.



# Ramar / Frames

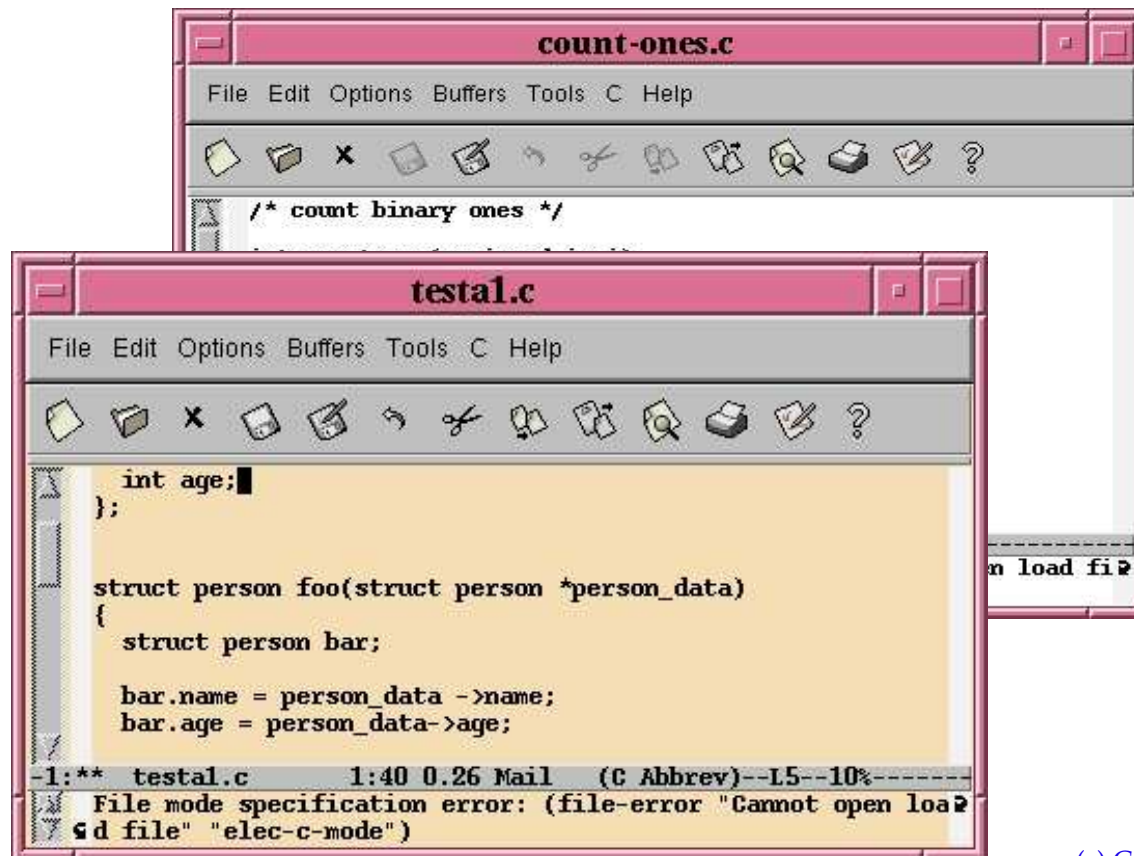
---

- I emacs benämns helt nya separata fönster frames.
- En ny ram / frame startas med C-x 5 2
- En ram / frame kan avslutas med C-x 5 0



# Ramar / Frames

- Exempel på två ramar / frames.



# Hjälp

---

- C-h ger inledningen till den inbyggda hjälpen i emacs.
- C-h ? ger ett fönster med den hjälp som går att få.
- Scrolla i listan med <mellanslag>, <delete>
- Samma lista går även att få med C-h C-h
- För att få hjälp om den mode som används för en buffert går det att trycka C-h m



# Hjälp

---

- C-h k ger hjälp om ett tangentkommando.
- Exempel: C-h k C-f ger hjälp om C-f.
- Det går även att få hjälp under Help i menyraden.



# Hjälp

---

- C-h f används för att hitta och beskriva en funktion.
- Exempel: C-h f forward-char
- För att hitta kortkommandot för en funktion kan aproposfunktionen användas. C-h a
- Exempel: C-h a forward ger alla funktioner som flyttar framåt och deras kortkommandon.
- Till C-h a kan ett reguljärt uttryck anges.



# Hjälp

---

- C-h b listar alla tangentbordsbindningar som är aktiva nu.
- I emacs finns inbyggt funktion för komplettering med <tab> och i en del fall även <mellanslag>.
- När ett alternativ ska ges, t.ex. buffertnamn, kan man skriva ? för att få upp en lista med alla alternativ.



# Tutorial

---

- I emacs finns en inbyggd tutorial som nås med C-h t.
- Den är en bra introduktion till emacs och innehåller övningar.





# .emacs

---

```
; Slå av ljudlig bell och gå över till visible bell  
(setq-default visible-bell t)
```

```
; Slå på matchning av parenteser  
(show-paren-mode 1)
```

```
; Byt bakgrundsfärg  
(set-background-color "wheat")
```

```
; Fast markör istället för blinkande  
(blink-cursor-mode nil)
```



# .emacs

---

```
; Exempel på att läsa in en annan fil.  
;; Läs in erlang-mode  
(and (file-readable-p "~/elisp/erlang.el")  
      (load "~/elisp/erlang.el"))  
  
; Läs in python-mode  
(and (file-readable-p "~/elisp/python-mode.el")  
      (autoload 'python-mode "~/elisp/python-mode" nil t))  
  
; Visa tid i mode-raden  
(display-time)
```



# Ändra utseende på emacs

---

- Ändra och ställa in utseende på emacs kan bland annat göras från menyn under Options->Customize Emacs



# Typsnitt

---

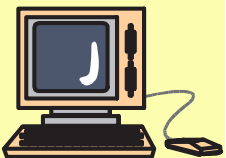
- Typsnitt går att ange på kommandoraden när man startar emacs med -fn (se tidigare bild).
- Det går även att byta typsnitt i emacs med <shift>-vänster musknapp.
- Denna ger en typsnittsmeny med några val.



# Programutveckling

---

- Det finns moder till nästan alla språk.
- När en källkodsfil läses in ställer emacs automatiskt in sig i rätt mode (oftast).
- Det går att ändra mode med: M-x språk-mode
- T.ex. M-x java-mode



# Programutveckling

---

- Kompilera program från emacs kan vanligtvis göras med C-x C-k eller C-x c
- Fyll på med namnet på programmet.
- Blir t.ex. make -k hello
- Programmet kompileras och felmeddelanden och utmatning visas i en egen buffert.
- För att gå till nästa fel går det att göra: M-x next error eller C-x '
- För att gå till en specifik rad kan man göra: M-x goto-line



# Programutveckling

---

- Emacs moder för olika språk hanterar automatiskt indentering.
- Om man läser in eller kopierar in någon annans kod så kanske den är felindenterad. Markera en region och tryck M-x indent-region för att indentera regionen.
- På liknande sätt går det att kommentera ut en hel region med kod med M-x comment-region.
- En region med bortkommenterad kod kan avkommenteras med M-x uncomment-region.



# Programutveckling

---

- ett vanligt paket för färgning av kod är hilit. Använd följande i .emacs för att få det:  
(load "hilit19")
- M-/ kompletterar namn på t.ex. en variabel eller funktion.
- Skriv början på namnet och tryck sedan på M-/ .





# Programutveckling

---

- Det går att checka in och ut filer från t.ex. cvs direkt i emacs.
- C-x vv
- C-x C-q



# Debugging med gdb

---

- Program går att debugga från emacs. Kompilera med flaggan -g.
- Starta gdb från emacs med M-x gdb. Ange att emacs ska köra gdb som gdb programnam.
- Sätt en breakpoint i main. Nu går det att stega fram i programmet med step eller s.
- Genom att visa källkoden i ett eget fönster ser man hur debuggern stegar fram i programmet.



# Utskrifter

---

- För att få snygga utskrifter av t.ex. programkod går det att göra M-x ps-print-buffer
- ps-print ger postscript-utskrift.
- För att skriva till en fil går det att göra C-u M-x ps-print-buffer följt av sökväg och filnamn.



# LysKOM

---

- Dokumentation till LysKOM finns på:  
<http://www.lysator.liu.se/lyskom>
- En introduktion till elisp-klienten finns på:  
<http://www.lysator.liu.se/~kjell-e/tekla/linux/dokument/>
- LysKOM-klienter finns på:  
<ftp://ftp.lysator.liu.se/pub/lyskom/elisp-client>

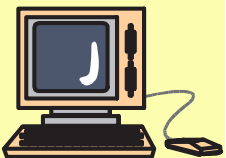


# .emacs och LysKOM

---

```
; Ladda in LysKOM
(autoload 'lyskom "~/lyskom-swedish-0.48.elc"
          "Run LysKOM" t)

(setq kom-default-server "kom.lysator.liu.se")
(setq kom-default-user-name "dittnamn")
(setq kom-default-language 'sv)
```



# Övrigt

---

- För att få snygg radbrytning på ett stycke går det att göra genom att markera stycket och sedan trycka M-x fill-region.
- Radbrytningen beror på inställningar av radlängd.
- Ett kommando kan upprepas med C-u.
- T.ex. för att gå 4 tecken framåt C-u C-f.
- för att upprepa ett annat antal än 4 ange antal till C-u.
- T.ex. C-u 20 # ger 20 stycken #-tecken.



# Övrigt

---

- Uppprepning går även att få med M-<tal>.
- Det finns en stavningskontroll som startas med M-x ispell.
- Ange den bokstav eller siffra för det alternativ som texten ska ersättas med.
- Slå ? för att få upp hjälp.



# Övrigt roligt

---

- Det finns en kalender som man kan komma åt med M-x calendar.
- I emacs finns det inbyggt en psykolog (eliza) som kan nås med M-x doctor.
- Det finns även ett program som kan prata med doktorn :-) Detta får man fram med M-x psychoanalyze-pinhead





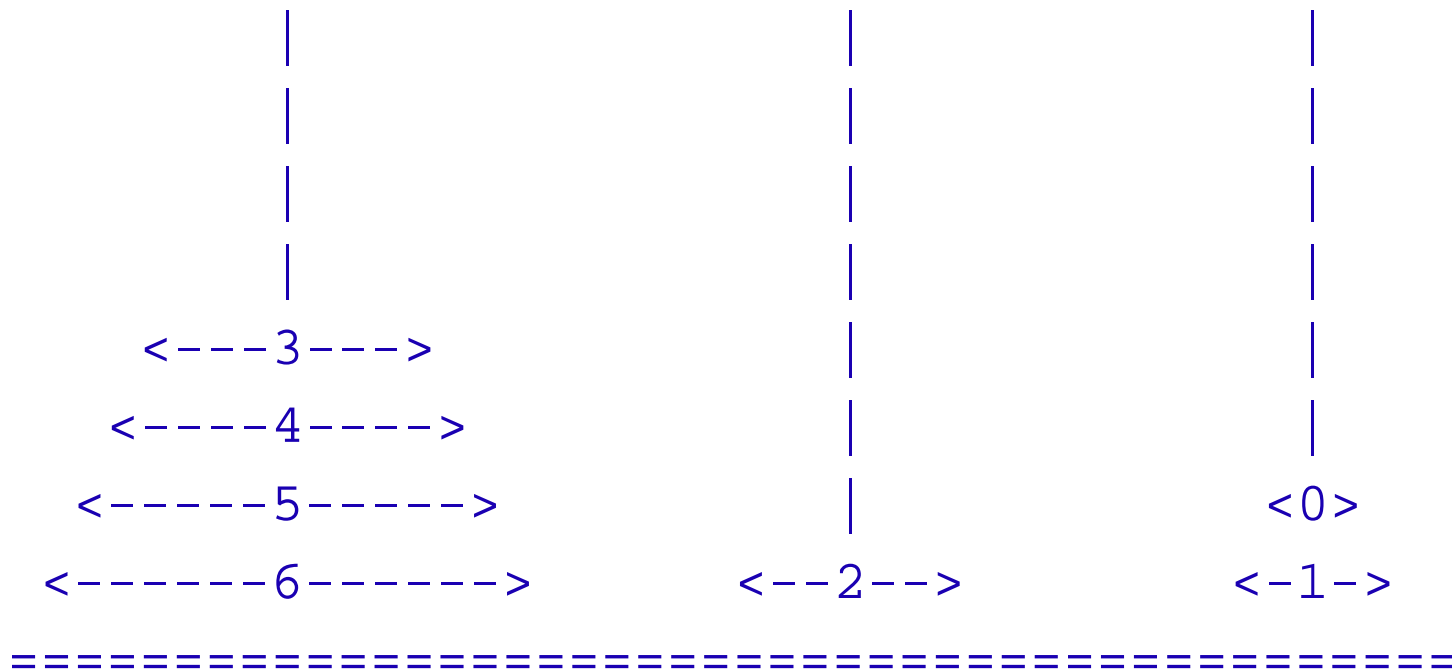
# Övrigt roligt

---

- Tetris går att starta med M-x tetris. Använd sedan piltangenterna.
- Den inbyggda Towers of Hanoi går att nå med M-x hanoi.
- Om man vill ha större torn, t.ex. 7 i storlek, gör man C-u 7 M-x hanoi



# Towers of Hanoi



# Tangentbordsmakron

---

- Det går att spela in makron.
- C-x ( startar inspelningen av ett makro.
- C-x ) avslutar inspelningen av ett makro.
- C-x e kör det senaste makrot.
- M-x name-last-kbd-macro namnger senaste makrot.
- Ett namngivet makro kan sedan nås med M-x makrot.



# Tangentbordsmakron

---

- Ett namngivet makro kan sparas.
- M-x insert-kbd-macro  
<return>Makronamn<return>  
klistrar in makrot som anges.
- För att spara makrot i .emacs gör enligt följande: Gör klart makrot och ge det ett namn. Läs in .emacs och klistra in makrot enligt ovan i .emacs.
- Nu är makrot sparad och kan användas när helst man vill.



# Bytekompilera elisp-filer

---

- Elispfiler kan bytekompileras för att få exekveringen effektivare och snabbare.
- En elispfil bytekompileras med:  
M-x byte-compile-file
- Ange sökväg och filnamn.
- Nu finns en fil med ändelsen .elc som kan användas istället.
- Ändra till att ladda in bytekompilerade .elc-filer i .emacs istället för .el-filer.

