Being Productive With Emacs

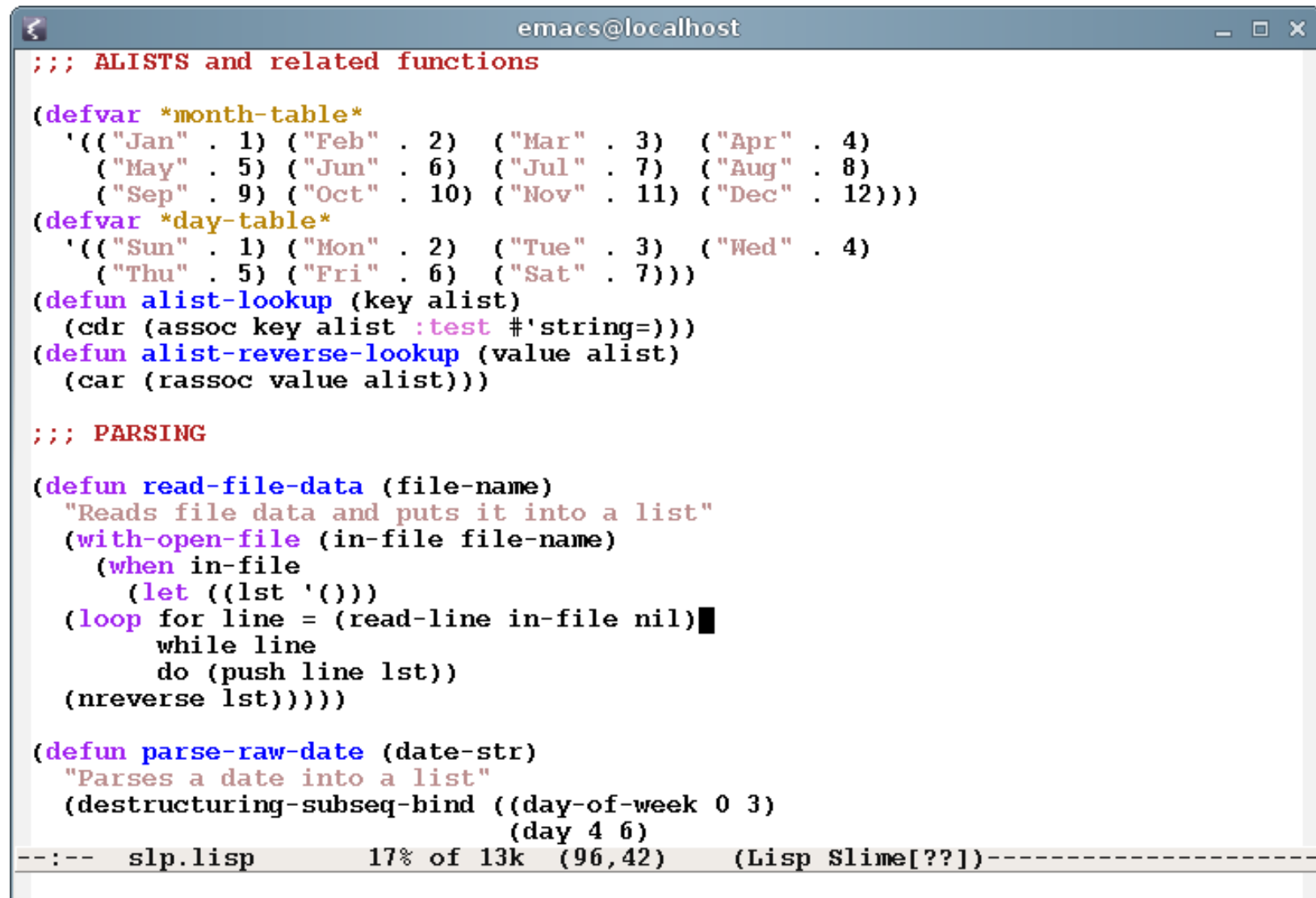## Part 1

Phil Sung

`sipb-iap-emacs@mit.edu`

`http://stuff.mit.edu/iap/emacs`

Special thanks to Piaw Na and Arthur Gleckler

"**Emacs** is the extensible, customizable, self-documenting real-time display editor."

The many faces of Emacs

```
;;; ALISTS and related functions

(defvar *month-table*
  '(("Jan" . 1) ("Feb" . 2)  ("Mar" . 3)  ("Apr" . 4)
    ("May" . 5) ("Jun" . 6)  ("Jul" . 7)  ("Aug" . 8)
    ("Sep" . 9) ("Oct" . 10) ("Nov" . 11) ("Dec" . 12))))
(defvar *day-table*
  '(("Sun" . 1) ("Mon" . 2)  ("Tue" . 3)  ("Wed" . 4)
    ("Thu" . 5) ("Fri" . 6)  ("Sat" . 7))))
(defun alist-lookup (key alist)
  (cdr (assoc key alist :test #'string=)))
(defun alist-reverse-lookup (value alist)
  (car (rassoc value alist)))

;;; PARSING

(defun read-file-data (file-name)
  "Reads file data and puts it into a list"
  (with-open-file (in-file file-name)
    (when in-file
      (let ((lst '()))
  (loop for line = (read-line in-file nil)█
        while line
        do (push line lst))
  (nreverse lst)))))

(defun parse-raw-date (date-str)
  "Parses a date into a list"
  (destructuring-subseq-bind ((day-of-week 0 3)
                              (day 4 6)
```
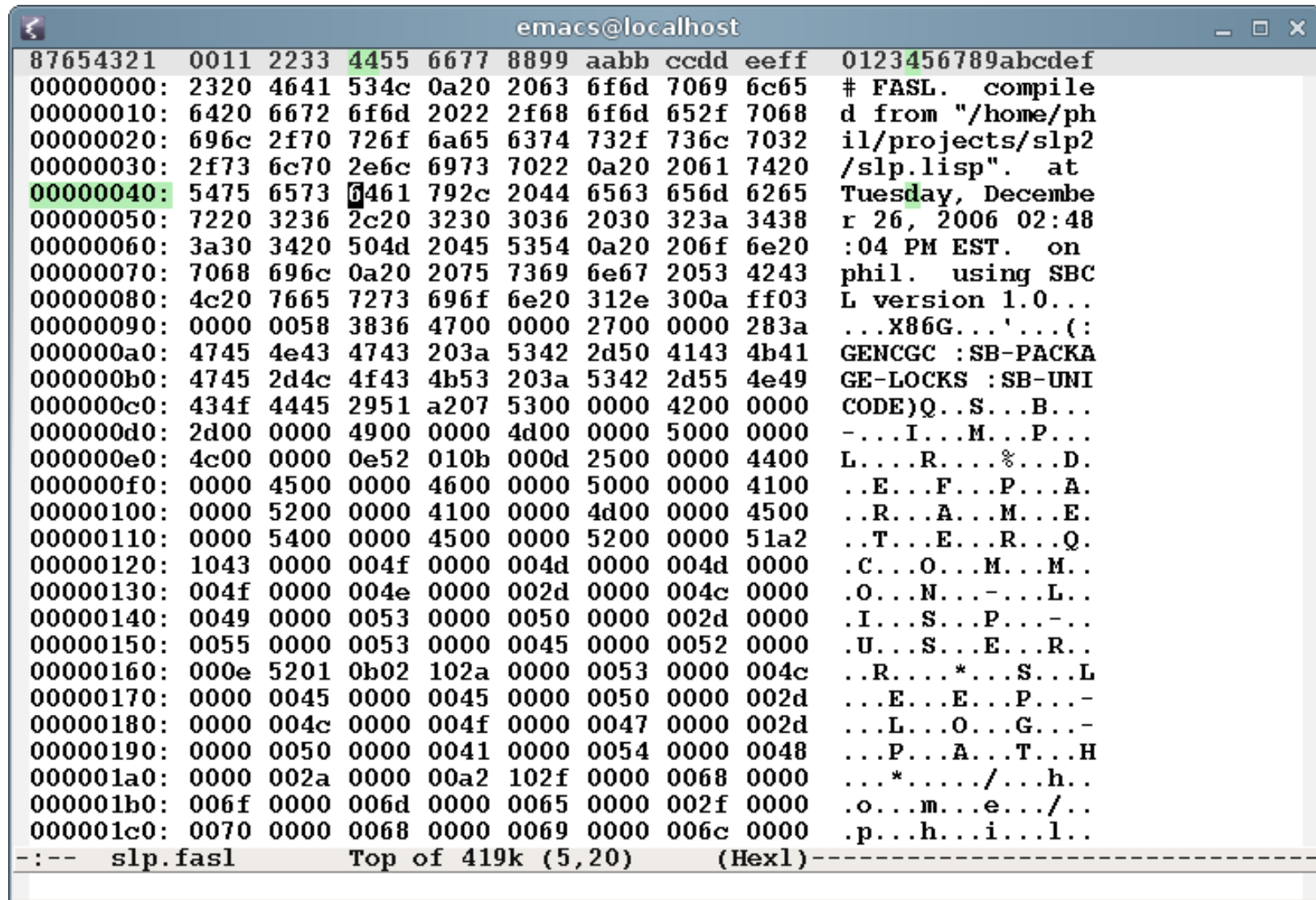
--:-- slp.lisp   17% of 13k (96,42)   (Lisp Slime[??])---------------------

Emacs edits source code

The many faces of Emacs


```
87654321   0011 2233 4455 6677 8899 aabb ccdd eeff   0123456789abcdef
00000000:  2320 4641 534c 0a20 2063 6f6d 7069 6c65   # FASL.  compile
00000010:  6420 6672 6f6d 2022 2f68 6f6d 652f 7068   d from "/home/ph
00000020:  696c 2f70 726f 6a65 6374 732f 736c 7032   il/projects/slp2
00000030:  2f73 6c70 2e6c 6973 7022 0a20 2061 7420   /slp.lisp".  at
00000040:  5475 6573 6461 792c 2044 6563 656d 6265   Tuesday, Decembe
00000050:  7220 3236 2c20 3230 3036 2030 323a 3438   r 26, 2006 02:48
00000060:  3a30 3420 504d 2045 5354 0a20 206f 6e20   :04 PM EST.  on
00000070:  7068 696c 0a20 2075 7369 6e67 2053 4243   phil.  using SBC
00000080:  4c20 7665 7273 696f 6e20 312e 300a ff03   L version 1.0...
00000090:  0000 0058 3836 4700 0000 2700 0000 283a   ...X86G...'...(:
000000a0:  4745 4e43 4743 203a 5342 2d50 4143 4b41   GENCGC :SB-PACKA
000000b0:  4745 2d4c 4f43 4b53 203a 5342 2d55 4e49   GE-LOCKS :SB-UNI
000000c0:  434f 4445 2951 a207 5300 0000 4200 0000   CODE)Q..S...B...
000000d0:  2d00 0000 4900 0000 4d00 0000 5000 0000   -...I...M...P...
000000e0:  4c00 0000 0e52 010b 000d 2500 0000 4400   L....R....%...D.
000000f0:  0000 4500 0000 4600 0000 5000 0000 4100   ..E...F...P...A.
00000100:  0000 5200 0000 4100 0000 4d00 0000 4500   ..R...A...M...E.
00000110:  0000 5400 0000 4500 0000 5200 0000 51a2   ..T...E...R...Q.
00000120:  1043 0000 004f 0000 004d 0000 004d 0000   .C...O...M...M..
00000130:  004f 0000 004e 0000 002d 0000 004c 0000   .O...N...-...L..
00000140:  0049 0000 0053 0000 0050 0000 002d 0000   .I...S...P...-..
00000150:  0055 0000 0053 0000 0045 0000 0052 0000   .U...S...E...R..
00000160:  000e 5201 0b02 102a 0000 0053 0000 004c   ..R....*...S...L
00000170:  0000 0045 0000 0045 0000 0050 0000 002d   ...E...E...P...-
00000180:  0000 004c 0000 004f 0000 0047 0000 002d   ...L...O...G...-
00000190:  0000 0050 0000 0041 0000 0054 0000 0048   ...P...A...T...H
000001a0:  0000 002a 0000 00a2 102f 0000 0068 0000   ...*...../...h..
000001b0:  006f 0000 006d 0000 0065 0000 002f 0000   .o...m...e.../..
000001c0:  0070 0000 0068 0000 0069 0000 006c 0000   .p...h...i...l..
-:--   slp.fasl        Top of 419k (5,20)       (Hexl)----------------------------
```

Emacs is a hex editor

`M-x hexl-find-file`

The many faces of Emacs

Emacs does diffs

`M-x ediff-buffers`

The many faces of Emacs

Emacs is a file manager

`M-x dired`

The many faces of Emacs

```
emacs@localhost                                              _ □ ✕
"An idealist is one who, on noticing that a rose smells better than a
cabbage, concludes that it will also make better soup." - H.L. Mencken
phil@phil ~ $ cat /proc/cpuinfo | head
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 15
model           : 2
model name      : Intel(R) Pentium(R) 4 CPU 2.80GHz
stepping        : 9
cpu MHz         : 2793.163
cache size      : 512 KB
physical id     : 0
siblings        : 1
phil@phil ~ $ ps -e | wc -l
93
phil@phil ~ $ █




-u:**  *shell*          All of 420  (16,14)      (Shell:run)----------------------
```

Emacs is a shell

`M-x shell`

The many faces of Emacs

Emacs is a mail/news client

`M-x gnus`

The many faces of Emacs
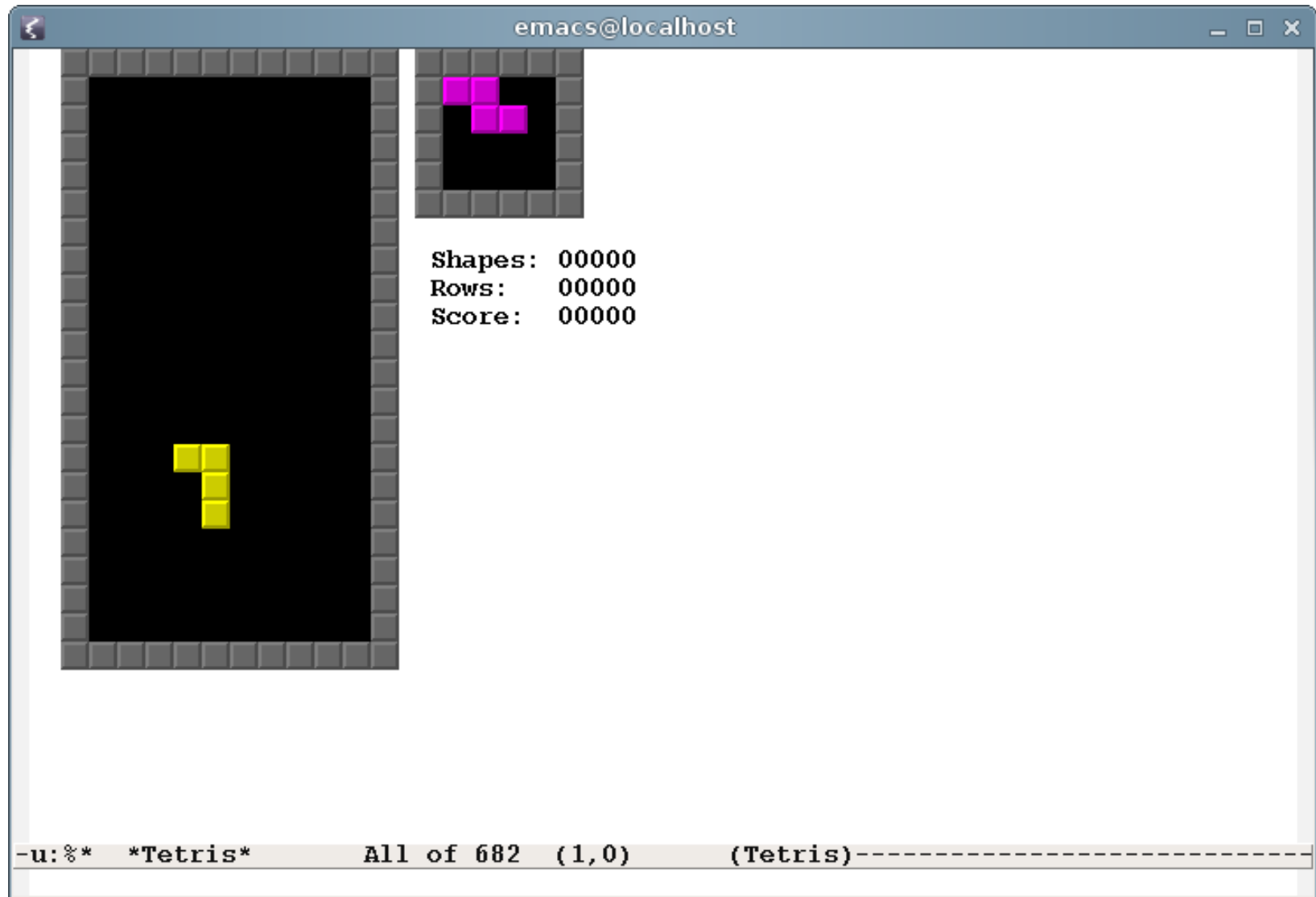
Emacs plays tetris

`M-x tetris`

Why Emacs?

- Provides an integrated environment

  - Same editing commands available everywhere

  - Large set of tools available at all times

  - Move text between tasks easily

Why Emacs?

- Easy to extend

  - Elisp for customizing or adding new features

  - Extension code has the full power of Emacs

  - Dynamic environment: no restarting or recompiling

- Portable

- **Today's goal: get the flavor of Emacs**

  - Getting started with Emacs

  - Editing tips

  - Demos of useful features

  - Common Emacs concepts

- **Later...**

  - Advanced customization

  - Programming and extending Emacs with Elisp

Examples based on
GNU Emacs 22

Prerequisites (sort of)

- Emacs basic concepts

  - Files, buffers, windows, frames

- Keyboard commands

  - Key commands, prefix keys, `M-x`, the minibuffer

  - `"C-x"` means **Ctrl+x**

    `"M-x"` means **Meta+x** or **Alt+x**

- Basic tasks

  - Opening and saving files, exiting Emacs

It's all about text manipulation

- **Text in files**

  - grocery lists, HTML, code, ...

- **Text outside of files**

  - shell, debugger, ...

- **Text as a metaphor**

  - dired, gnus, ...

Text as a metaphor: dired

```
emacs@localhost.locald
/home/phil/projects/emacs-talk:
total used in directory 296 available 28869812
drwxr-xr-x  3 phil phil    4096 2007-02-17 17:54 .
drwxr-xr-x 12 phil phil    4096 2007-03-23 15:06 ..
-rw-r--r--  1 phil phil    8408 2007-02-17 17:54 emacs-slides
-rw-r--r--  1 phil phil  199108 2007-03-07 22:29 emacs-slides-1.odp
-rw-r--r--  1 phil phil   33740 2007-02-17 18:37 emacs-slides-2.odp
-rw-r--r--  1 phil phil   33101 2007-02-17 18:37 emacs-slides-3.odp
drwxr-xr-x  2 phil phil    4096 2007-02-17 17:54 screens
```

```
-u:--    emacs-talk          All of 516   (7,50)      (Editable Dired)
```

After editing names in this buffer, `C-x C-s` renames the modified files

Moving around in buffers

- By character or line
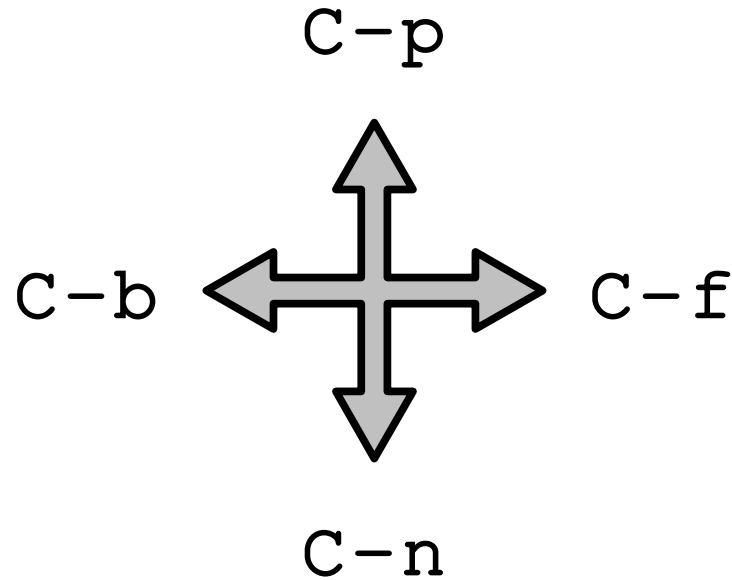
Moving around in buffers

- Beginning, end of line
  - `C-a, C-e`

- By word
  - `M-f, M-b`

- By sentence
  - `M-a, M-e`

- By screen
  - `C-v, M-v`

- Beginning, end of buffer
  - `M-<, M->`

- Go to line #
  - `M-g g`

Moving around in buffers

- Move multiple lines forward, backward

  - Example: `C-u 10 C-p` (back 10 lines)

  - `C-u` prefix generalizes to other commands

- Search for text

  - `C-s, C-r`

- Exchange point (cursor) and mark

  - `C-x C-x`

Killing ("cutting") text

- Kill line
  - `C-k`

- Kill many lines
  - `C-u 10 C-k` (10 lines)
  - `C-u C-k` (4 lines)
  - `C-u C-u C-k` (16 lines)

Killing ("cutting") text

- Kill region
  - `C-w`

- Save without killing
  - `M-w`

- Kill sentence
  - `M-k`

- Kill ("zap") to next occurrence of character
  - `M-z CHAR`

Yanking ("pasting") text

- Yank

  - `C-y`

- Yank earlier killed text

  - `M-y` (once or more after `C-y`)

- The kill ring

  - Almost all commands which delete text save it for possible later retrieval

The mark

- Remembers a previous cursor position
  - `C-x C-x` to swap point (cursor) and mark

- When you...
  - `C-spc`
  - `M-<` or `M->`
  - Search for text
  - Yank text
  - Insert a buffer

- the mark is set to...
  - where you are
  - where you were
  - where you started
  - start of inserted text
  - start of inserted text

The mark

- The mark ring
  - Move to a previous mark: `C-u C-SPC`

- Mark and point are also used to delineate 'the region'
  - Many commands operate on the text in the region
  - Set region by setting mark, then moving point

Undo

- Undo previous actions
  - `C-/` or `C-_` or `C-x u`

- Undo within current region
  - `C-u C-/`
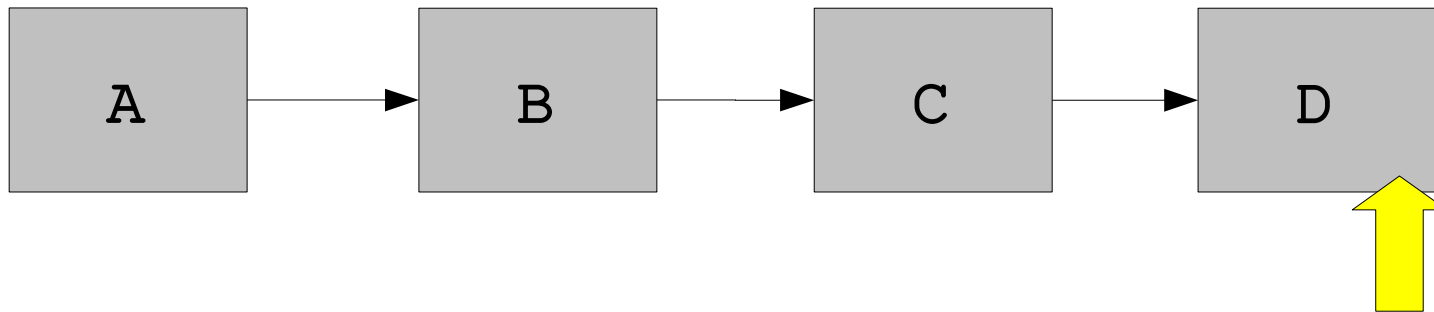
The undo model, illustrated

The undo model, illustrated

These states are accessible with 'undo'
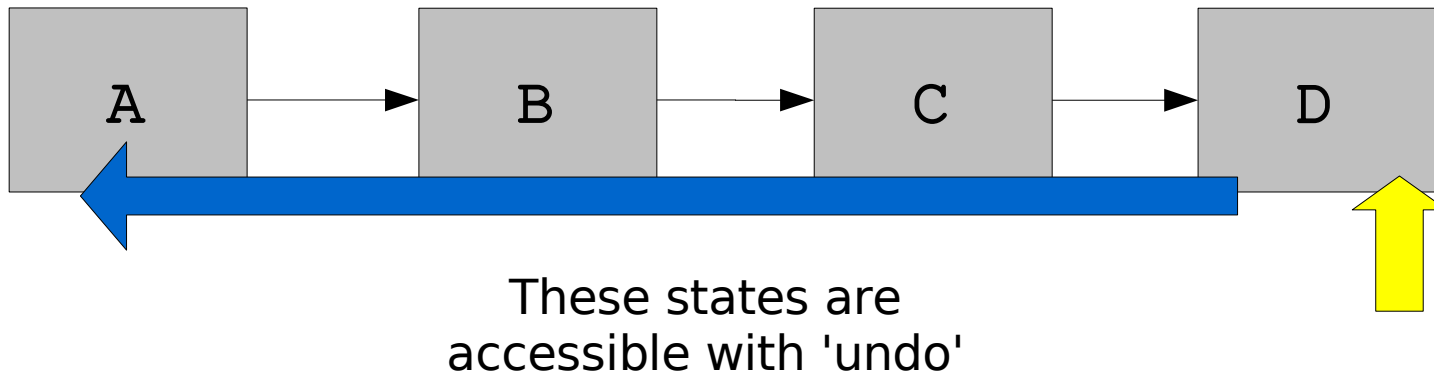
The undo model, illustrated

Undo some of your actions...

This is how most editors other than emacs work:

These states are
accessible with 'undo'

These states are
accessible with 'redo'
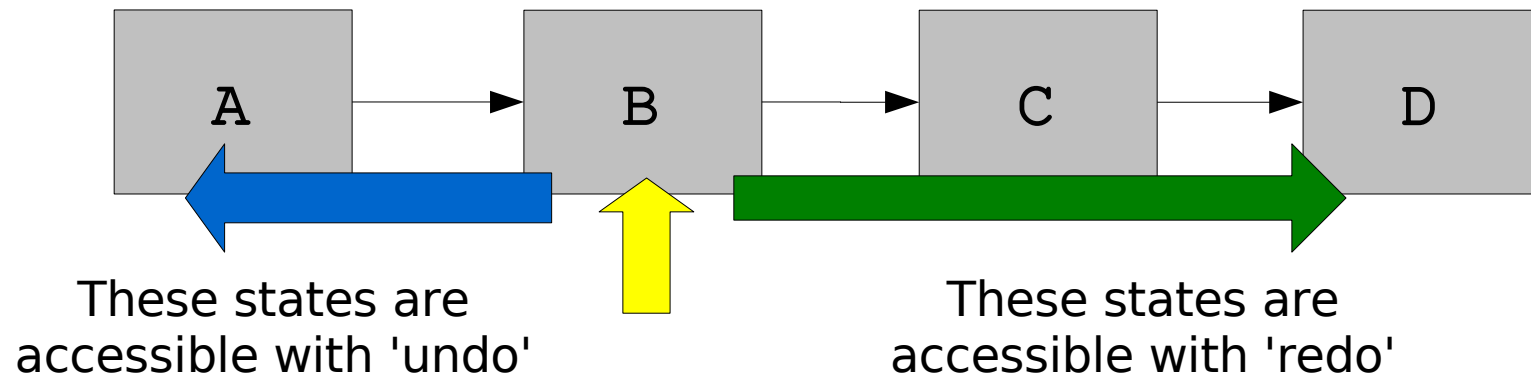
The undo model, illustrated

Now do something else...

This is how most editors other than emacs work:

These states are
accessible with 'undo'

These states are no
longer accessible!
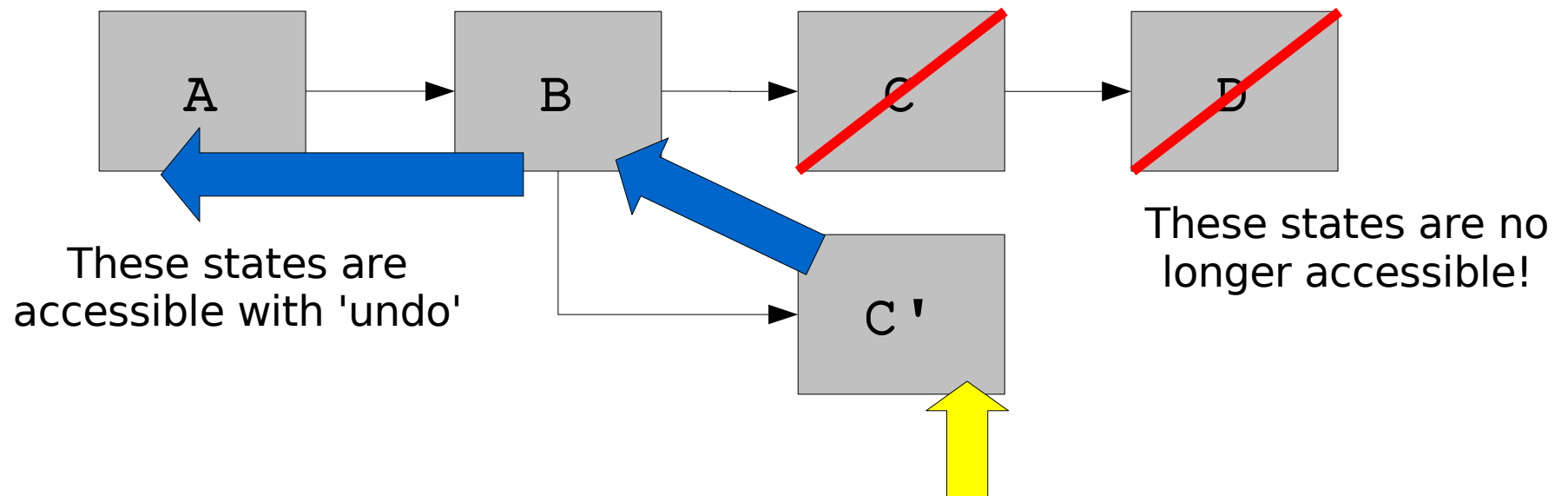
The undo model, illustrated

How emacs handles this situation
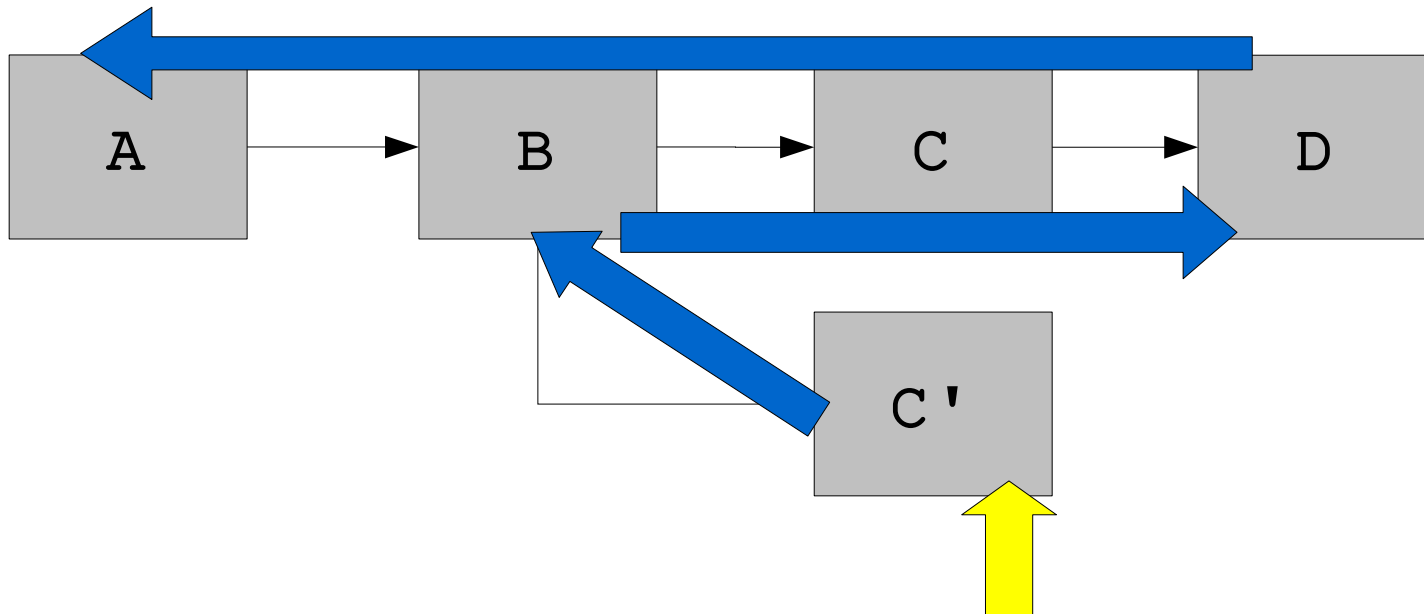
The list of states is 'folded' so that all previous actions, including undos, are undoable

Incremental search

- Search for text (like Firefox's "find as you type")
  - `C-s` *text*
    - `C-s` again to find next occurrence
    - `RET` to stop at found occurrence
    - `C-g` to cancel and go back to start of search
  - `C-r` for reverse search
  - Many options available inside search;
    `C-h k C-s` to learn more

Search history

- Search for previously searched string

  - `C-s  C-s`

- Browse and edit previous queries

  - `C-s` then `M-p`, `M-n`

Incremental search

- Search for regular expressions
  - `C-M-s` *regexp*
  - Regexp describes the form of what to look for
  - Syntax may be slightly different from other REs you may have used
    - Emacs REs are a superset of Perl REs
  - `M-x re-builder` can help you test complex regexps

Searching and replacing

- Search and replace, asking for confirmation

  - `M-%` or `M-x query-replace`

- Display all lines matching RE

  - `M-x occur`

RE search and replacement

- `M-x replace-regexp`

- Replacement text can depend on found text!

- Replacement text gets these substitutions:

  - `\&` (the matched string)

  - `\1`, `\2`, etc. (references to parts of matched string)

  - `\#` (number of matched lines so far)

  - `\?` (prompt user for what to enter)

  - `\,(lisp-expression ...)`

RE replacement example

```
Bill Gates
Steve Jobs
Eric Schmidt
Larry Ellison
```
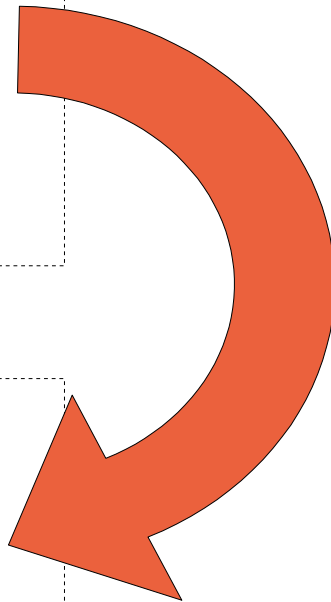
M-x replace-regexp
\(\w+\) \(\w+\)
with
\,(upcase \2), \1

```
GATES, Bill
JOBS, Steve
SCHMIDT, Eric
ELLISON, Larry
```

More tips at http://steve-yegge.blogspot.com/2006/06/shiny-and-new-emacs-22.html

Integration with useful tools

- Shell
  - `M-x shell`

- Compile (invoke make)
  - `M-x compile`

- Debug
  - `M-x gdb`

Integration with useful tools

- Grep
  - `M-x grep, M-x rgrep`

- Man page reader
  - `M-x woman`

- Invoke shell commands
  - `M-x shell-command,`
    `M-x shell-command-on-region`

...and then some

- Calculator
  - `M-x calc`

- Calendar
  - `M-x calendar`

- Moon calendar
  - `M-x phases-of-moon`

More helpful features

- TRAMP: open remote files over SSH

  - `C-x C-f /user@host:~/remote/file`

- VC: automatically deal with CVS, SVN, etc.

  - `M-x vc-next-action` to commit modified file

  - `M-x vc-diff` to view changes to current file

- etags: name search/completion for source code

Emacs server

- Use a single Emacs session for **all** editing

- Do this once: `M-x server-start`

  - or put `(server-start)` in your .emacs file

- To edit a file:

  - `prompt% emacsclient file`

  - File opens in existing Emacs frame

  - `C-x #` when done editing

Macros

- Remembers a fixed sequence of keys for later repetition

- Start recording macro: `C-x  (`

- Stop recording macro: `C-x  )`

- Replay macro: `C-x  e`

Macro example

Define macro:

```
M-d C-d M-u , [SPC]
C-y C-n C-a
```

"Remove first word and space, uppercase next word, insert comma and space afterward, reinsert first word, move to beginning of next line"

```
Bill Gates
Steve Jobs
Eric Schmidt
Larry Ellison
```
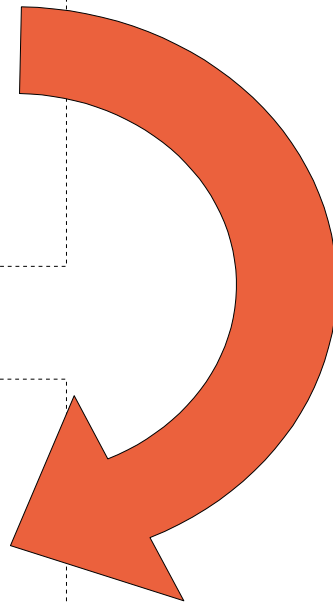
Run macro repeatedly:

```
C-x e e ...
```

```
GATES, Bill
JOBS, Steve
SCHMIDT, Eric
ELLISON, Larry
```

Narrowing

- Restricts view/editing in a buffer to a certain region

  - `C-x n n` or `M-x narrow-to-region` to narrow to region

  - `C-x n w` or `M-x widen` to restore ('widen')

Registers

- Store current window configuration

  - `C-x r w REGISTER`

- Restore window configuration

  - `C-x r j REGISTER`

- Registers can also store positions, text, numbers, file names...

REGISTER may be any letter or number

Prefix arguments

- Sometimes used to indicate repetition

  - `C-u 10 C-f` (forward 10 characters)

  - `C-u C-o` (make 4 new lines)

- Sometimes modify following command

  - `C-/` (undo) vs. `C-u C-/` (undo within region)

  - `M-x shell` vs. `C-u M-x shell`

- A command's documentation (`C-h f` or `C-h k`) describes the effect of the prefix argument, if any

Major modes

- Alters behavior, key bindings, and text display

- Switch mode in existing buffer:
  - `M-x java-mode`
  - `M-x python-mode`
  - `M-x fundamental-mode`

- Or, use another command to create buffer:
  - `M-x shell`
  - `M-x dired`

Language major mode features

- Language-specific indentation, syntax coloring

- Language-specific features:

  - Lisp: commands for manipulating s-expressions

  - Python: commands for (un)indenting blocks

  - HTML: insert/close tags; preview in web browser

  - Modes can define or redefine keys

Minor modes

- Extra functionality you can turn on or off

  - Any number of minor modes may be active at once

  - Independent of major mode functionality

- `M-x auto-fill-mode`

- `M-x flyspell-mode`

- `M-x follow-mode`

Global minor modes

- Offer completions for buffers, commands, etc.

  - `M-x icomplete-mode`

- Show all buffer names on `C-x b`:

  - `M-x iswitchb-mode`

Minibuffer input

- Common features whenever Emacs prompts you to enter something

  - Most buffer editing, movement commands work

  - Browse previous inputs with `M-n, M-p`

  - Tab-completion is often available

- `M-x eval-expression, M-x find-file, M-x switch-to-buffer,…`

Getting help with Emacs

- Help with key
  - `C-h k`

- Help with function
  - `C-h f`

- Help with mode
  - `C-h m`

- Show key bindings
  - `C-h b`

- Help about help
  - `C-h C-h`

Getting help with Emacs

- Apropos (search for command)

  - `C-h a`

- Help with prefix key

  - `C-h` (after prefix key)

- Manuals

  - `M-x info`, then select emacs or efaq

"In the event of an emergency"

- Cancel command
  - `C-g`
- `Undo!`
  - `C-/` or `C-_`
- What did I just do?
  - `M-x view-lossage`

Common problems

- Delete not deleting?

  - `M-x normal-erase-is-backspace-mode`

- Keys with `M-` not working?

  - Use `ESC` instead

  - `ESC x` instead of `M-x`

  - `ESC C-t` instead of `C-M-t`

Migrating to Emacs

- From Windows applications

  - `M-x cua-mode:` recovers `C-z, C-x, C-c, C-v` for their usual purposes

- From vi/vim

  - `M-x viper-mode`

Resources

- Emacs on Athena

    - http://web.mit.edu/olh/Emacs/

- Emacs reference card

    - http://web.mit.edu/olh/Emacs/Refcard.pdf

Invoking or installing Emacs

- **emacs21 on Athena**: `athena% emacs`

- **emacs22 on Ubuntu/Debian**:
  `apt-get install emacs-snapshot-gtk`

- **emacs22 on Gentoo**: `emerge emacs-cvs`

- **emacs on Windows**:

  - `http://ourcomments.org/Emacs/`
    `                EmacsW32Util.html`

- **Bonus: Being Unproductive With Emacs**

  - `M-x tetris`, `M-x hanoi`, `M-x doctor`, …

- **Next time: Emacs lisp**

  - Customizing how Emacs works

  - Writing new functions, commands, and modes

  - Manipulating text programmatically

  - Altering behavior of existing commands